

# Supplementary material

Optimisation of NMR dynamic models I. Minimisation algorithms  
and their performance within the model-free and Brownian  
rotational diffusion spaces.

by Edward J. d'Auvergne and Paul R. Gooley

## List of Figures

|     |                                                                                                                                                  |     |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| S1  | The $R_{ex}$ Grid $S^2$ difference surface for Modelfree4. . . . .                                                                               | S3  |
| S2  | The $R_{ex}$ Grid $S^2$ difference surface for Dasha. . . . .                                                                                    | S4  |
| S3  | The $R_{ex}$ Grid $S^2$ difference surface for relax. . . . .                                                                                    | S5  |
| S4  | The $R_{ex}$ Grid $\tau_e$ difference surface for Modelfree4. . . . .                                                                            | S6  |
| S5  | The $R_{ex}$ Grid $\tau_e$ difference surface for Dasha. . . . .                                                                                 | S7  |
| S6  | The $R_{ex}$ Grid $\tau_e$ difference surface for relax. . . . .                                                                                 | S8  |
| S7  | The $R_{ex}$ Grid $R_{ex}$ difference surface for Modelfree4. . . . .                                                                            | S9  |
| S8  | The $R_{ex}$ Grid $R_{ex}$ difference surface for Dasha. . . . .                                                                                 | S10 |
| S9  | The $R_{ex}$ Grid $R_{ex}$ difference surface for relax. . . . .                                                                                 | S11 |
| S10 | The Double Motion Grid $S^2$ difference surface for Modelfree4. . . . .                                                                          | S12 |
| S11 | The Double Motion Grid $S^2$ difference surface for Dasha. . . . .                                                                               | S13 |
| S12 | The Double Motion Grid $S^2$ difference surface for relax. . . . .                                                                               | S14 |
| S13 | The Double Motion Grid $S_f^2$ difference surface for Modelfree4. . . . .                                                                        | S15 |
| S14 | The Double Motion Grid $S_f^2$ difference surface for Dasha. . . . .                                                                             | S16 |
| S15 | The Double Motion Grid $S_f^2$ difference surface for relax. . . . .                                                                             | S17 |
| S16 | The Double Motion Grid $\tau_s$ difference surface for Modelfree4. . . . .                                                                       | S18 |
| S17 | The Double Motion Grid $\tau_s$ difference surface for Dasha. . . . .                                                                            | S19 |
| S18 | The Double Motion Grid $\tau_s$ difference surface for relax. . . . .                                                                            | S20 |
| S19 | Parameter deconvolution . . . . .                                                                                                                | S26 |
| S20 | Map of the model-free $\chi^2$ space of the grid point $S^2 = 0.831$ , $\tau_e = 256$ ps, and $R_{ex} = 1.644$ s <sup>-1</sup> S27               |     |
| S21 | Map of the model-free $\chi^2$ space of the grid point $S^2 = 0.376$ , $S_f^2 = 0.970$ , and $\tau_s = 4$ ps<br>( $S_s^2 = 0.388$ ) . . . . .    | S29 |
| S22 | Map of the model-free $\chi^2$ space of the grid point $S_f^2 = 0.952$ , $S_s^2 = 0.582$ , and $\tau_s = 2$ ps . . .                             | S32 |
| S23 | Map of the model-free $\chi^2$ space of the grid point $S^2 = 0.722$ , $S_f^2 = 0.931$ , and $\tau_s = 1024$ ps<br>( $S_s^2 = 0.776$ ) . . . . . | S34 |
| S24 | Map of the chi-squared space of the ellipsoid orientational diffusion tensor. . . . .                                                            | S38 |

## List of Tables

|     |                                                                                                                                                                                                                                  |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| S1  | Constrained optimisation of the grid point $S^2 = 0.831$ , $\tau_e = 256$ ps, and $R_{ex} = 1.644 \text{ s}^{-1}$ . .                                                                                                            | S28 |
| S2  | Comparison of the optimisation efficiency of various unconstrained optimisation algorithms for the extremely difficult to minimise grid point $S^2 = 0.376$ , $S_f^2 = 0.970$ , and $\tau_s = 4$ ps ( $S_s^2 = 0.388$ ). . . . . | S30 |
| S3  | Constrained optimisation algorithm comparison for the grid point $S^2 = 0.376$ , $S_f^2 = 0.970$ , and $\tau_s = 4$ ps ( $S_s^2 = 0.388$ ) . . . . .                                                                             | S31 |
| S4  | Constrained optimisation algorithm comparison for the grid point $S_f^2 = 0.952$ , $S_s^2 = 0.582$ , and $\tau_s = 2$ ps. . . . .                                                                                                | S33 |
| S5  | Constrained optimisation algorithm comparison for the grid point $S^2 = 0.722$ , $S_f^2 = 0.931$ , and $\tau_s = 1024$ ps ( $S_s^2 = 0.776$ ). . . . .                                                                           | S35 |
| S6  | Optimisation of the grid point $S^2 = 0.970$ , $\tau_e = 2048$ ps, and $R_{ex} = 0.149 \text{ s}^{-1}$ . . . . .                                                                                                                 | S36 |
| S7  | Optimisation of the grid point $S_f^2 = 0.952$ , $S_s^2 = 0.582$ , and $\tau_s = 32$ ps. . . . .                                                                                                                                 | S36 |
| S8  | High precision optimisation of the grid point $S_f^2 = 0.952$ , $S_s^2 = 0.582$ , and $\tau_s = 32$ ps after modification and recompilation of the Modelfree4 and Dasha source code. . . . .                                     | S36 |
| S9  | Optimisation of the grid point $S^2 = 0.388$ , $\tau_e = 128$ ps, and $R_{ex} = 0.223 \text{ s}^{-1}$ . . . . .                                                                                                                  | S37 |
| S10 | The parameter values of the optimised diffusion models of cytochrome $c_2$ . . . . .                                                                                                                                             | S37 |

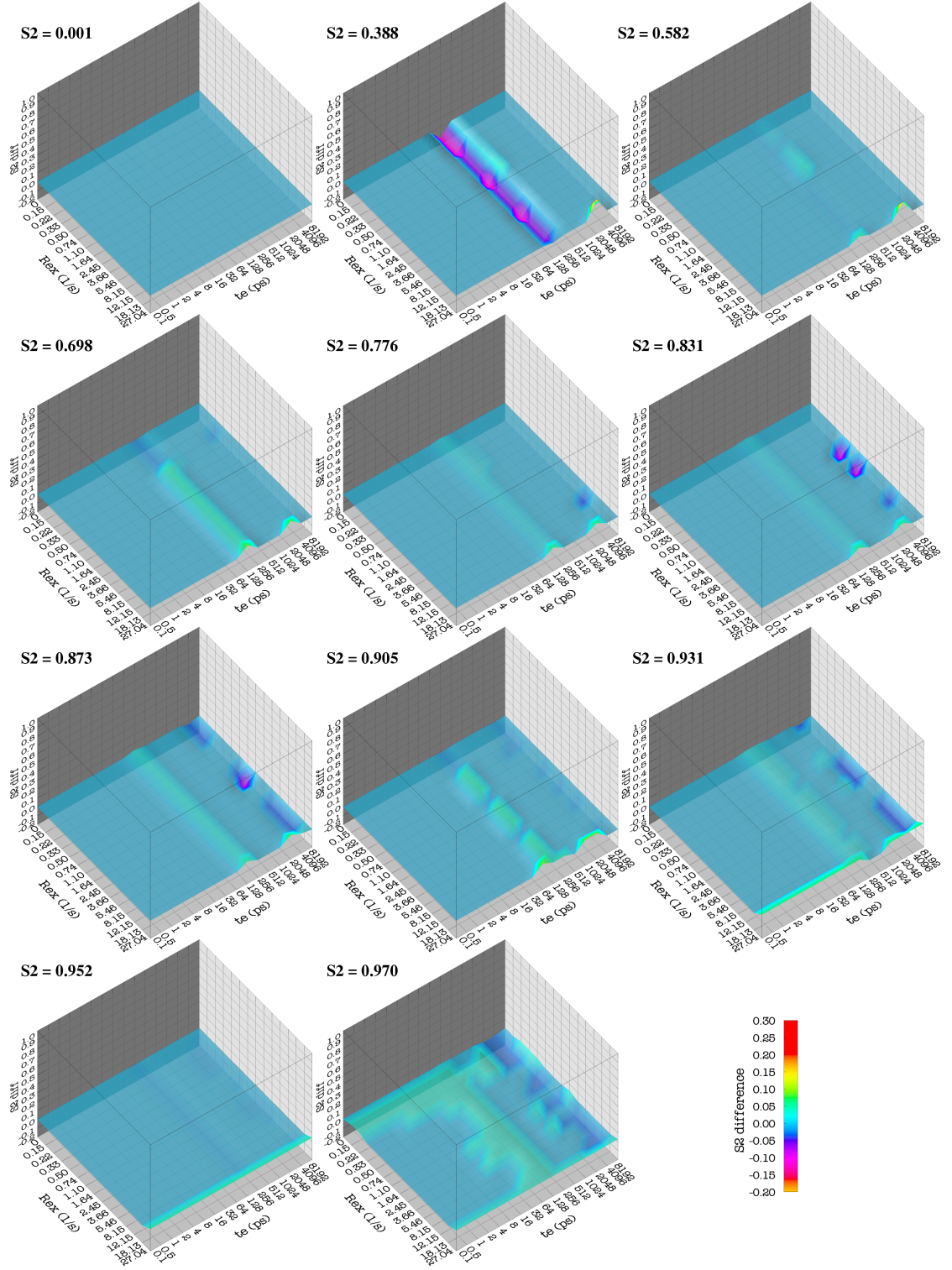


Figure S1: The results of the optimisation of the  $R_{\text{ex}}$  Grid using Modelfree4 presented as a  $S^2$  difference surface. For each grid point Modelfree4 was used to optimise  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.002.

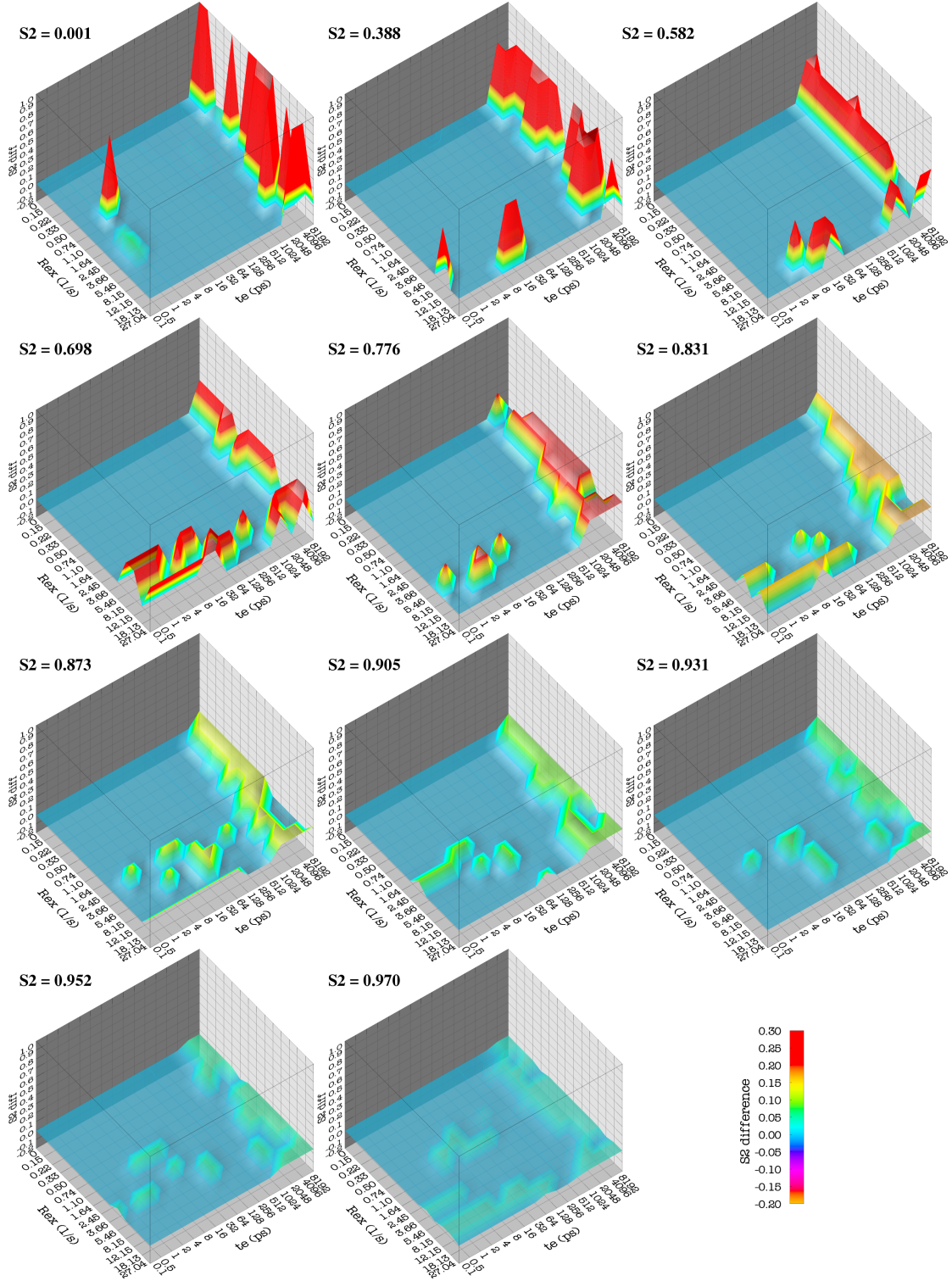


Figure S2: The results of the optimisation of the  $R_{ex}$  Grid using Dasha presented as a  $S^2$  difference surface. For each grid point Dasha was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.002.



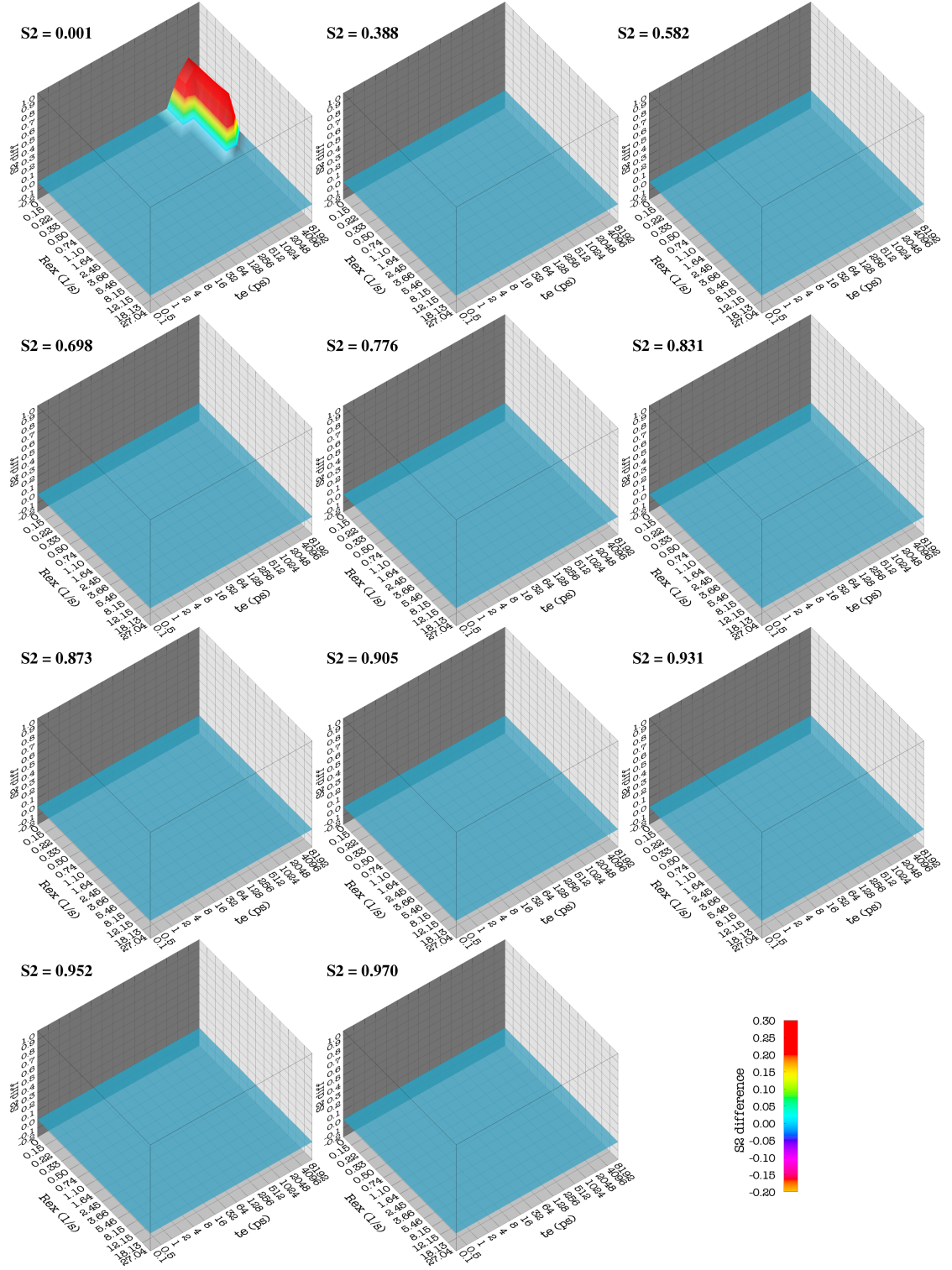


Figure S3: The results of the optimisation of the  $R_{ex}$  Grid using relax presented as a  $S^2$  difference surface. For each grid point relax was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.002.

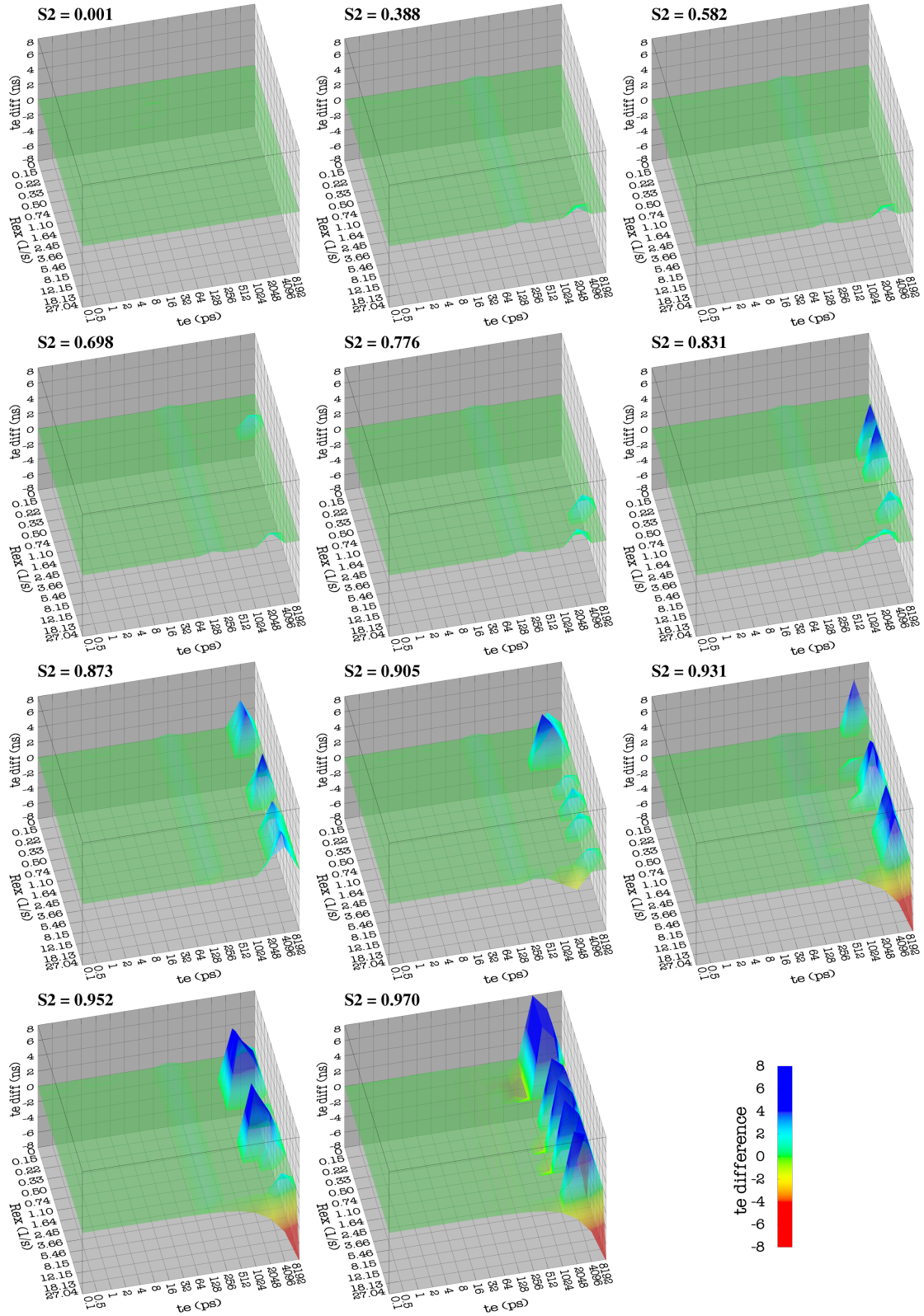


Figure S4: The results of the optimisation of the  $R_{ex}$  Grid using Modelfree4 presented as a  $\tau_e$  difference surface. For each grid point Modelfree4 was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.

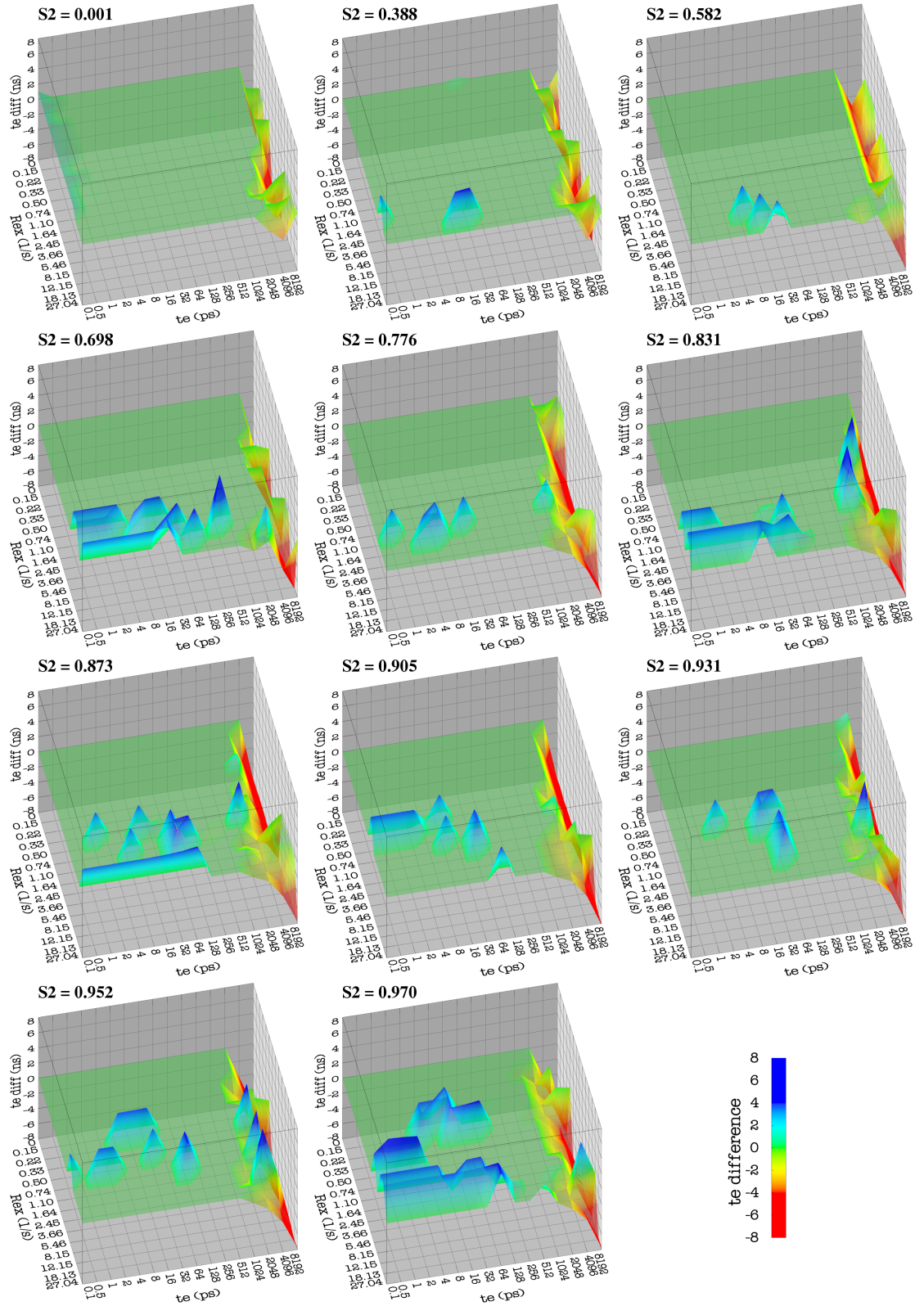


Figure S5: The results of the optimisation of the  $R_{ex}$  Grid using Dasha presented as a  $\tau_e$  difference surface. For each grid point Dasha was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.



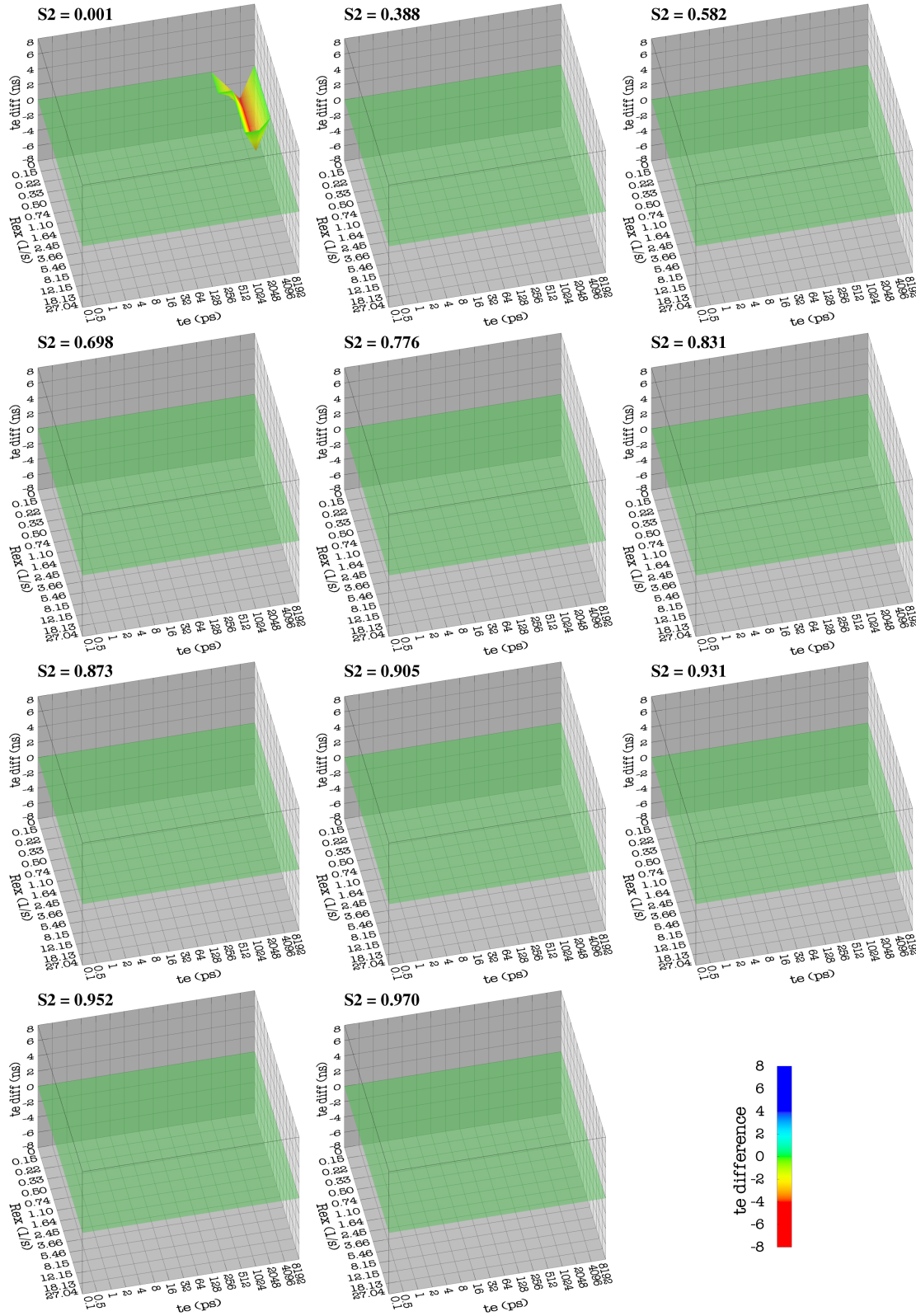


Figure S6: The results of the optimisation of the  $R_{ex}$  Grid using relax presented as a  $\tau_e$  difference surface. For each grid point relax was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.

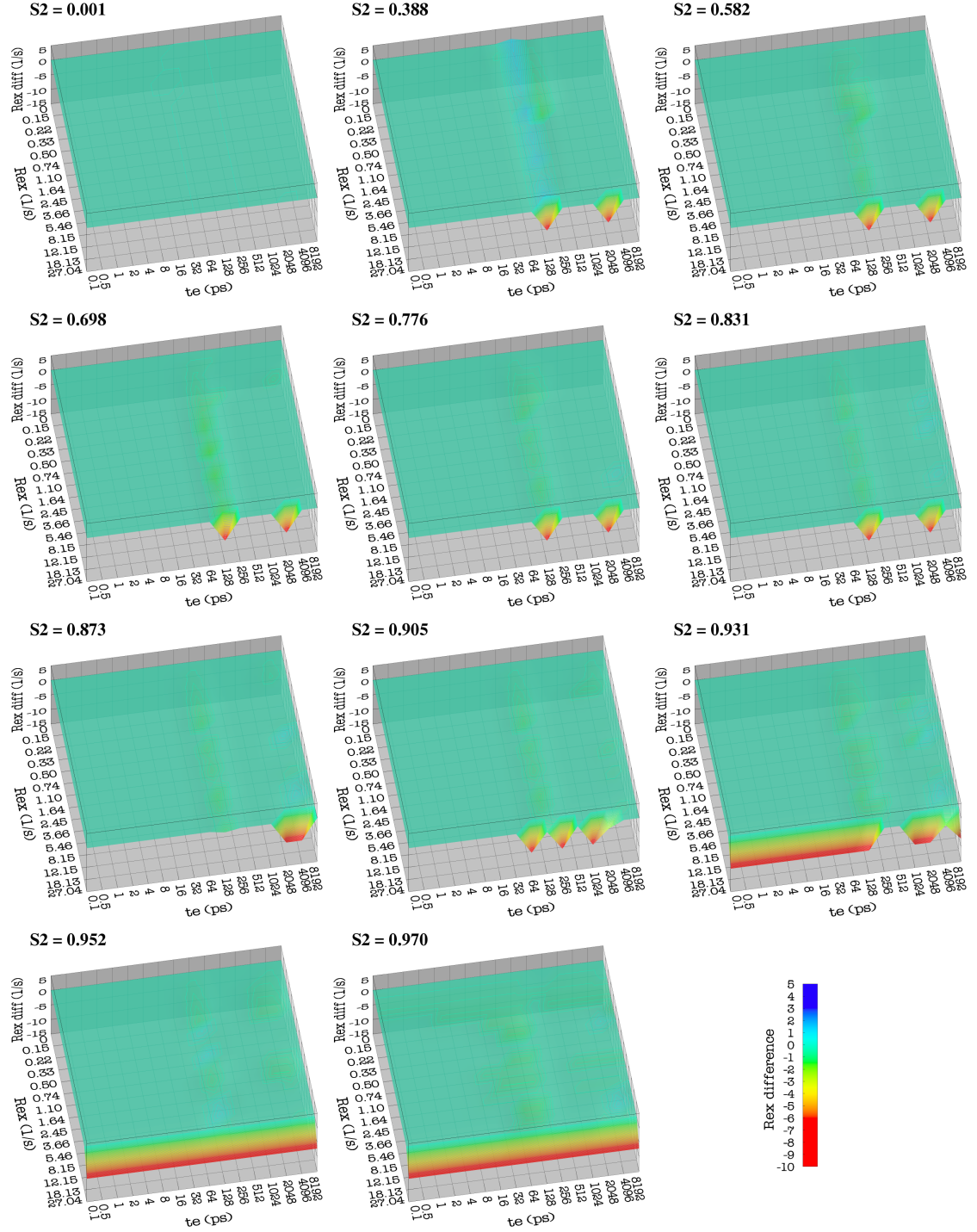


Figure S7: The results of the optimisation of the  $R_{ex}$  Grid using Modelfree4 presented as a  $R_{ex}$  difference surface. For each grid point Modelfree4 was used to optimise model  $m_4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is  $0.1 \text{ s}^{-1}$ .



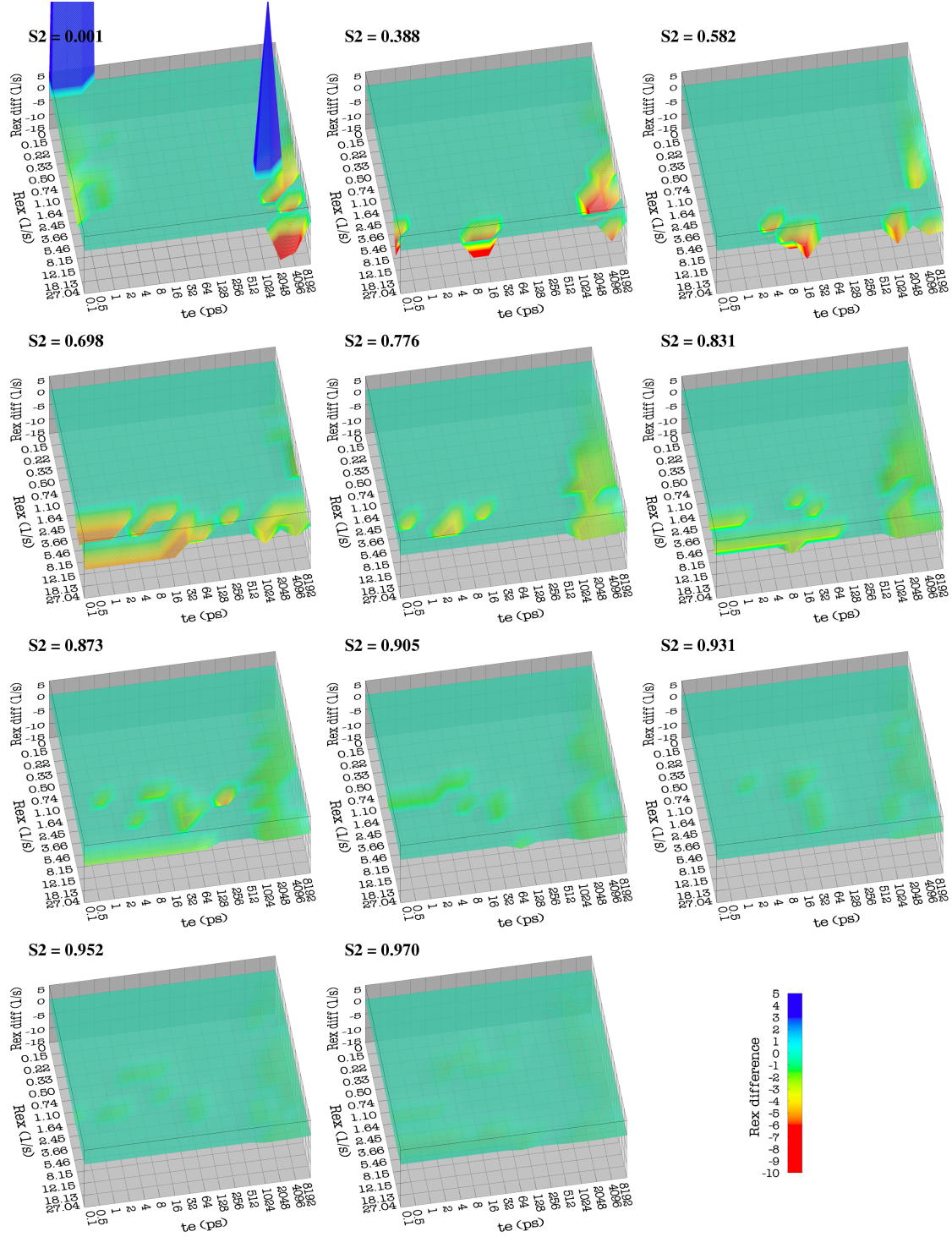


Figure S8: The results of the optimisation of the  $R_{ex}$  Grid using Dasha presented as a  $R_{ex}$  difference surface. For each grid point Dasha was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is  $0.1 \text{ s}^{-1}$ .

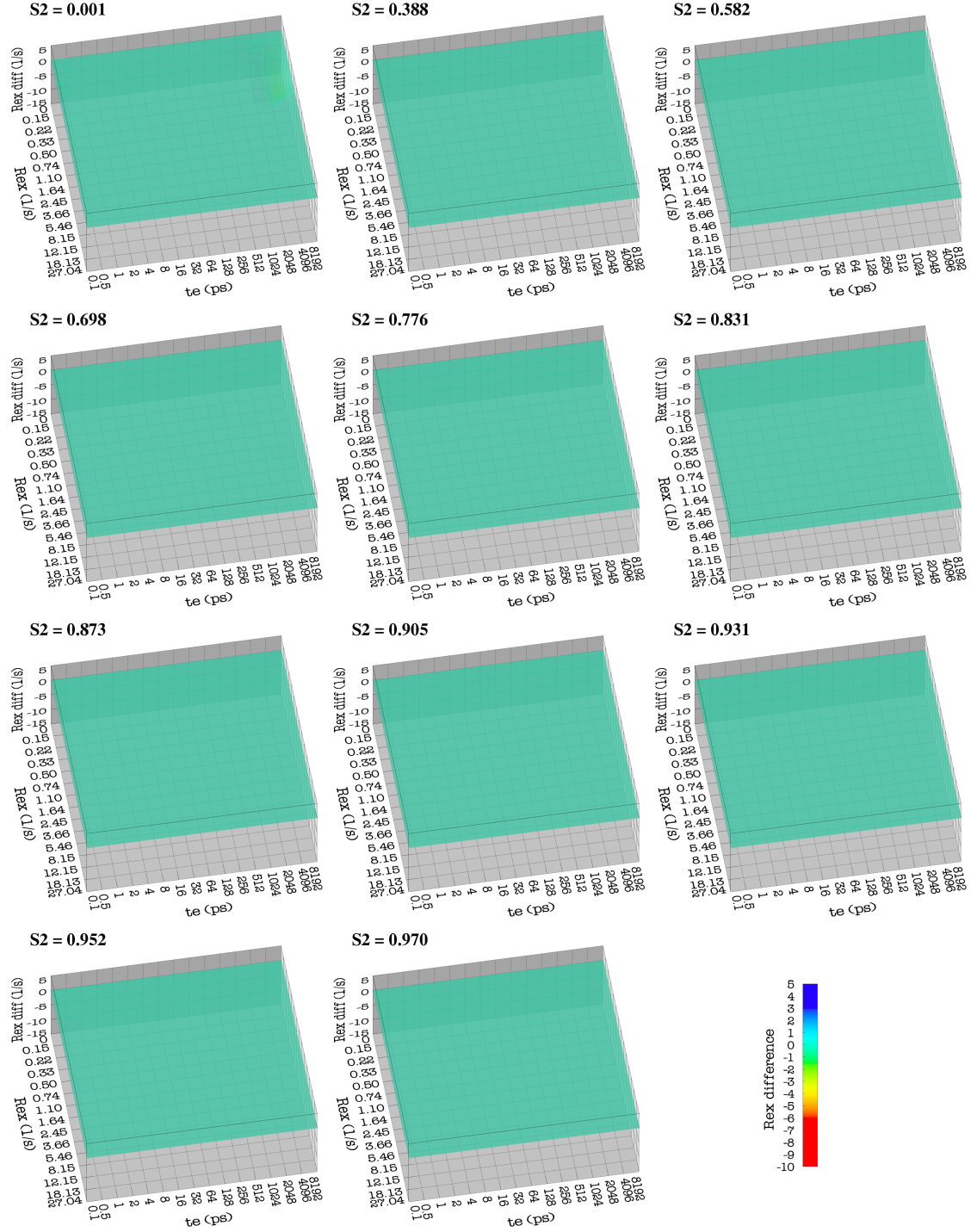


Figure S9: The results of the optimisation of the  $R_{ex}$  Grid using relax presented as a  $R_{ex}$  difference surface. For each grid point relax was used to optimise model  $m4$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is  $0.1 \text{ s}^{-1}$ .

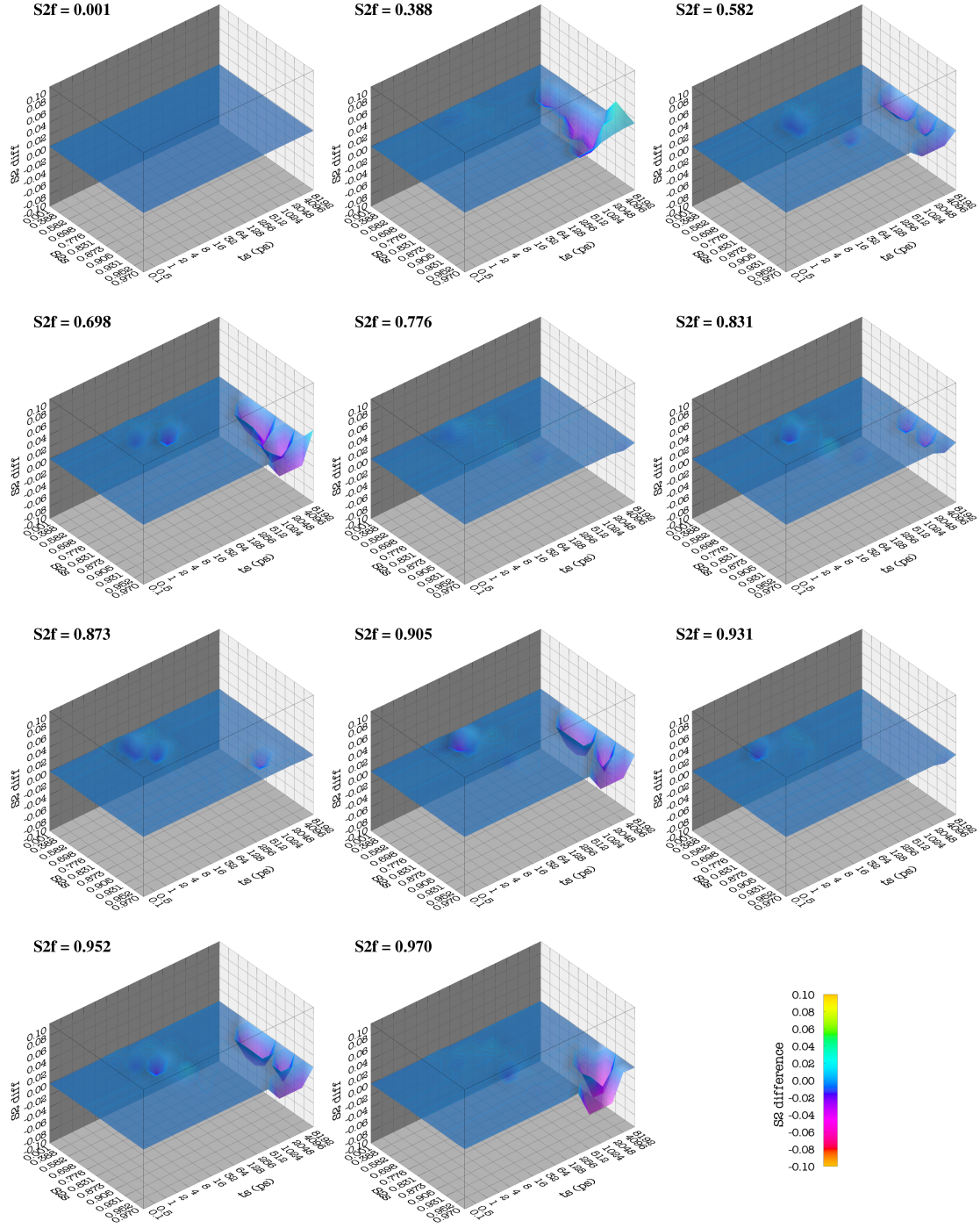


Figure S10: The results of the optimisation of the Double Motion Grid using Modelfree4 presented as a  $S^2$  difference surface. For each grid point Modelfree4 was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.001.

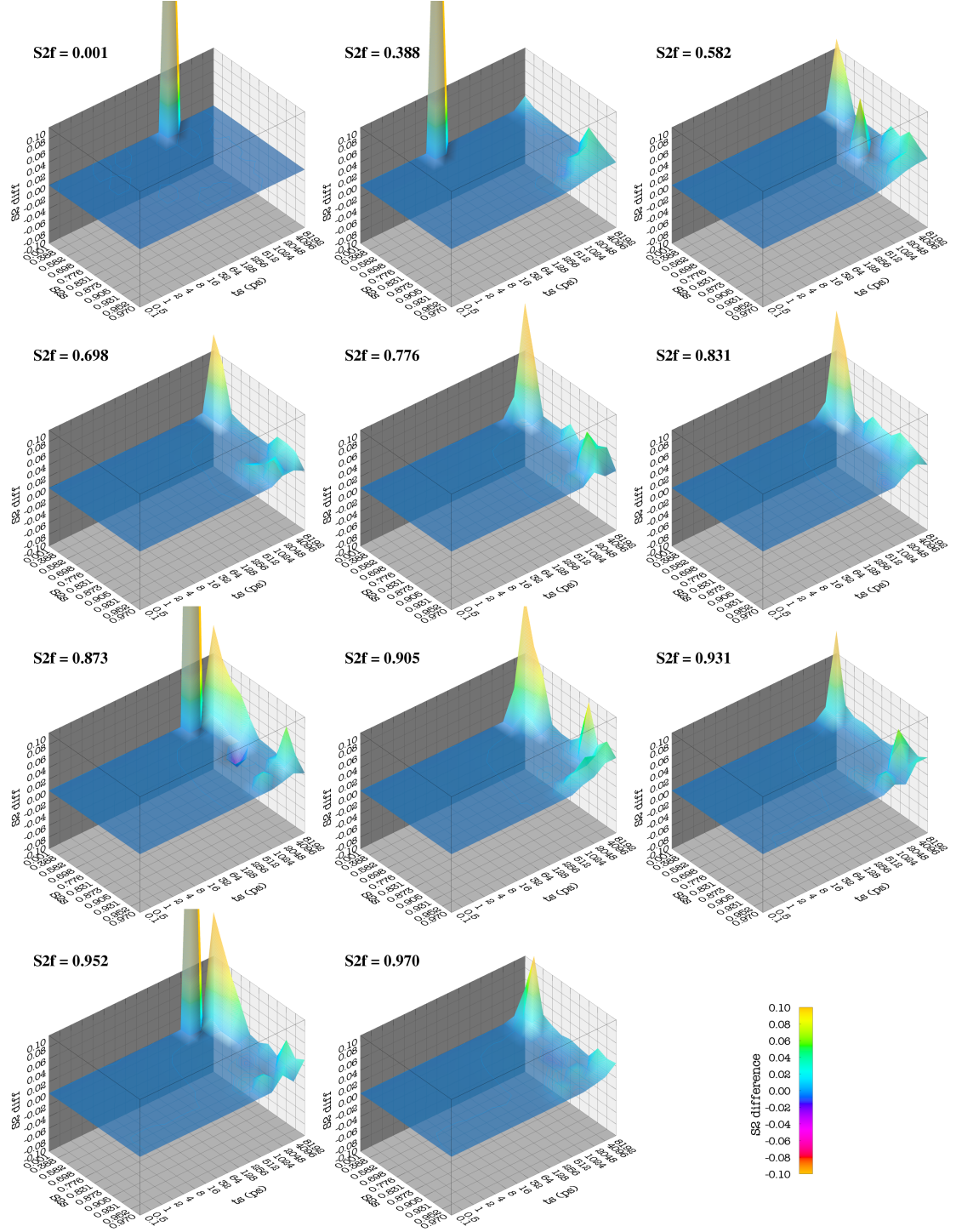


Figure S11: The results of the optimisation of the Double Motion Grid using Dasha presented as a  $S^2$  difference surface. For each grid point Dasha was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.001.



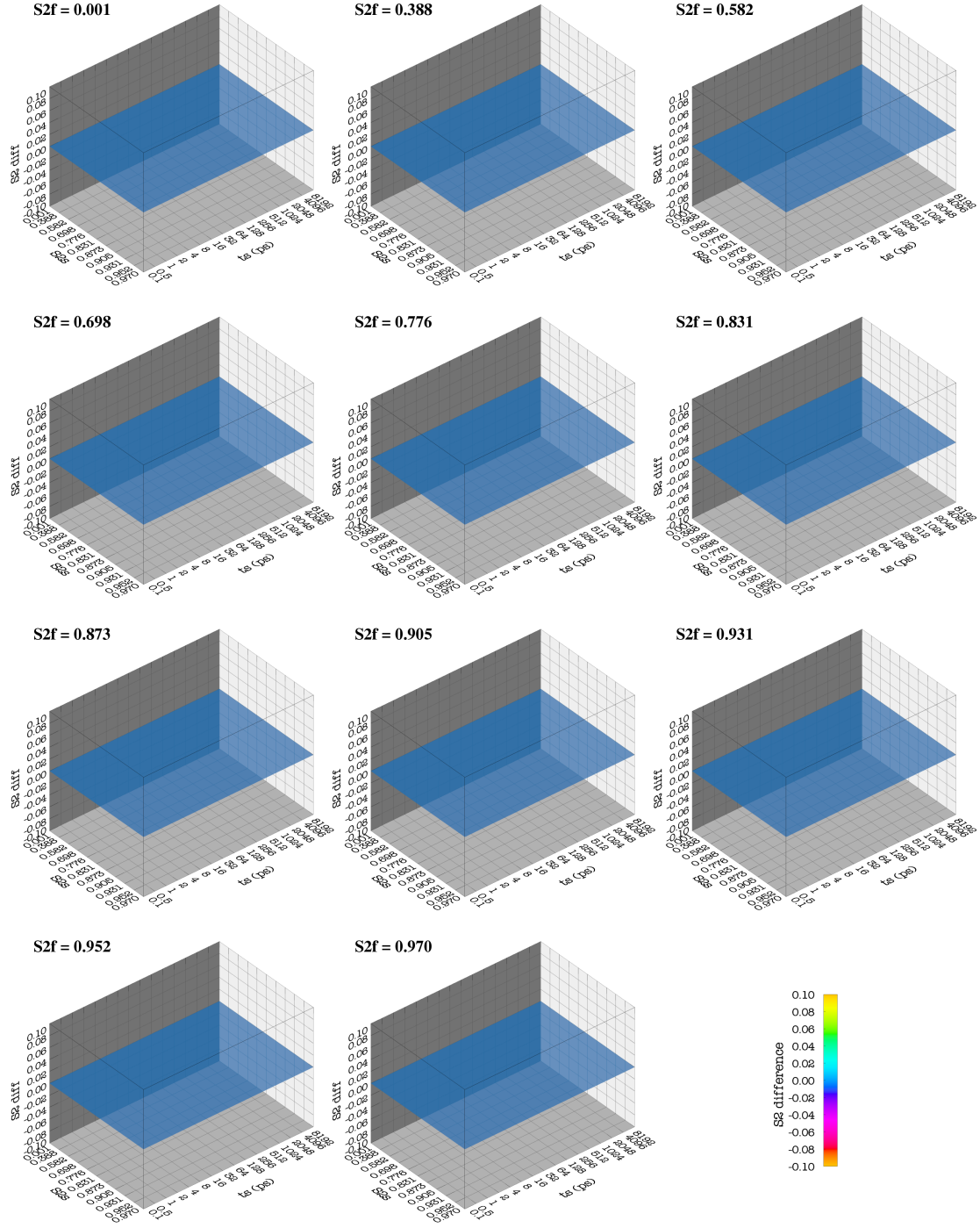


Figure S12: The results of the optimisation of the Double Motion Grid using relax presented as a  $S^2$  difference surface. For each grid point relax was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.001.

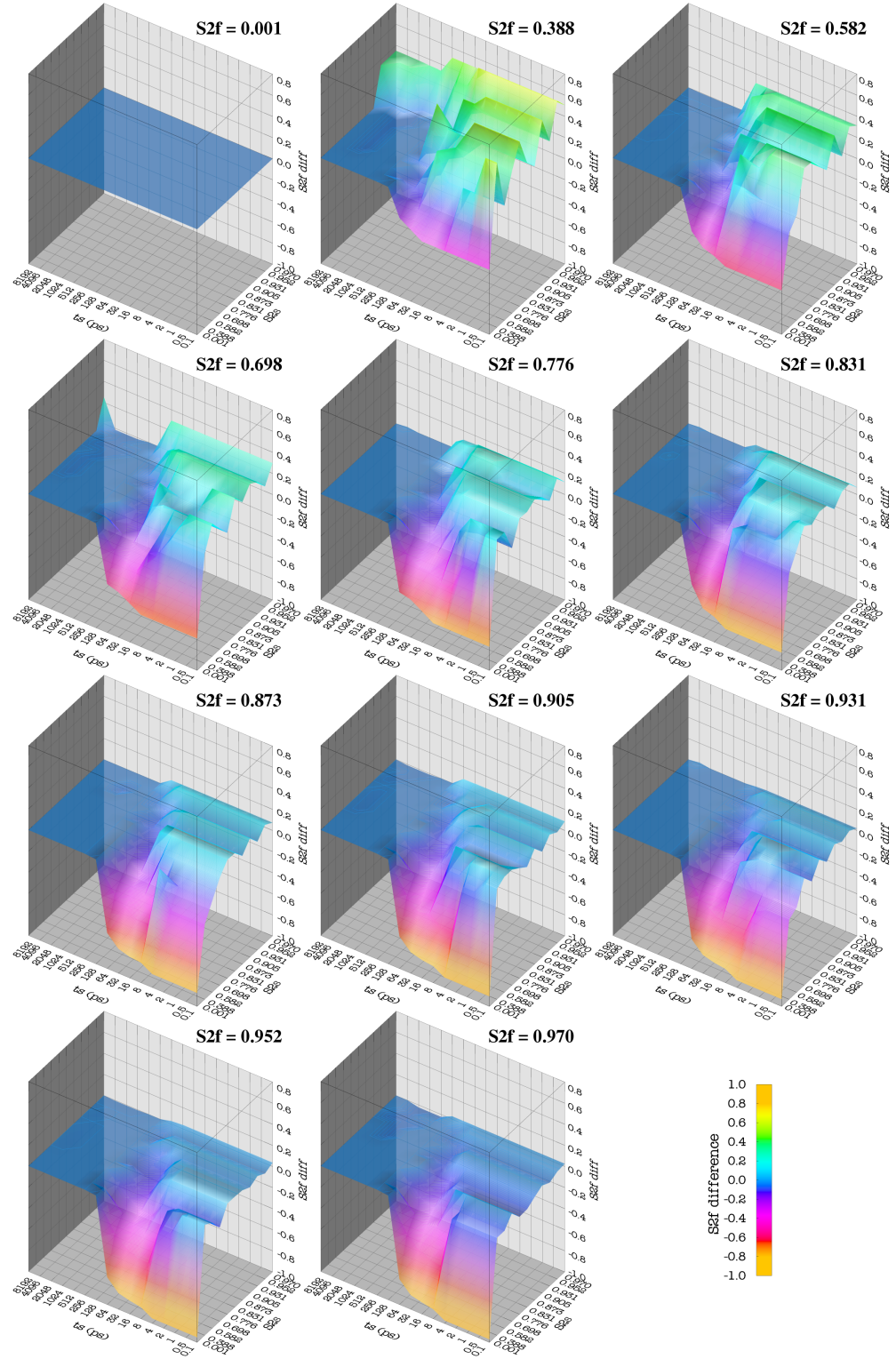


Figure S13: The results of the optimisation of the Double Motion Grid using Modelfree4 presented as a reverse angle view of the  $S_f^2$  difference surface. For each grid point Modelfree4 was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.005.

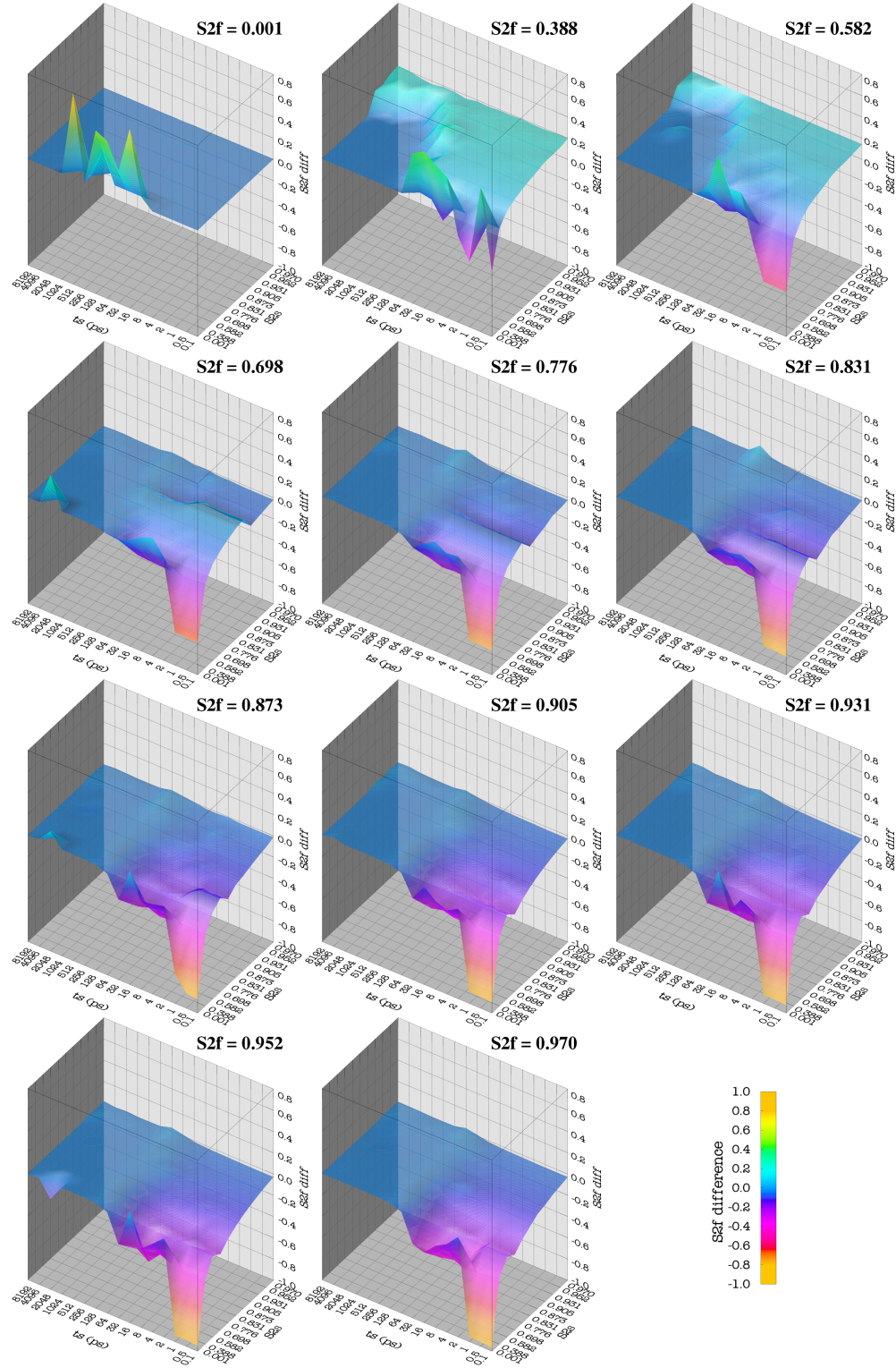


Figure S14: The results of the optimisation of the Double Motion Grid using Dasha presented as a reverse angle view of the  $S^2_f$  difference surface. For each grid point Dasha was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.005.



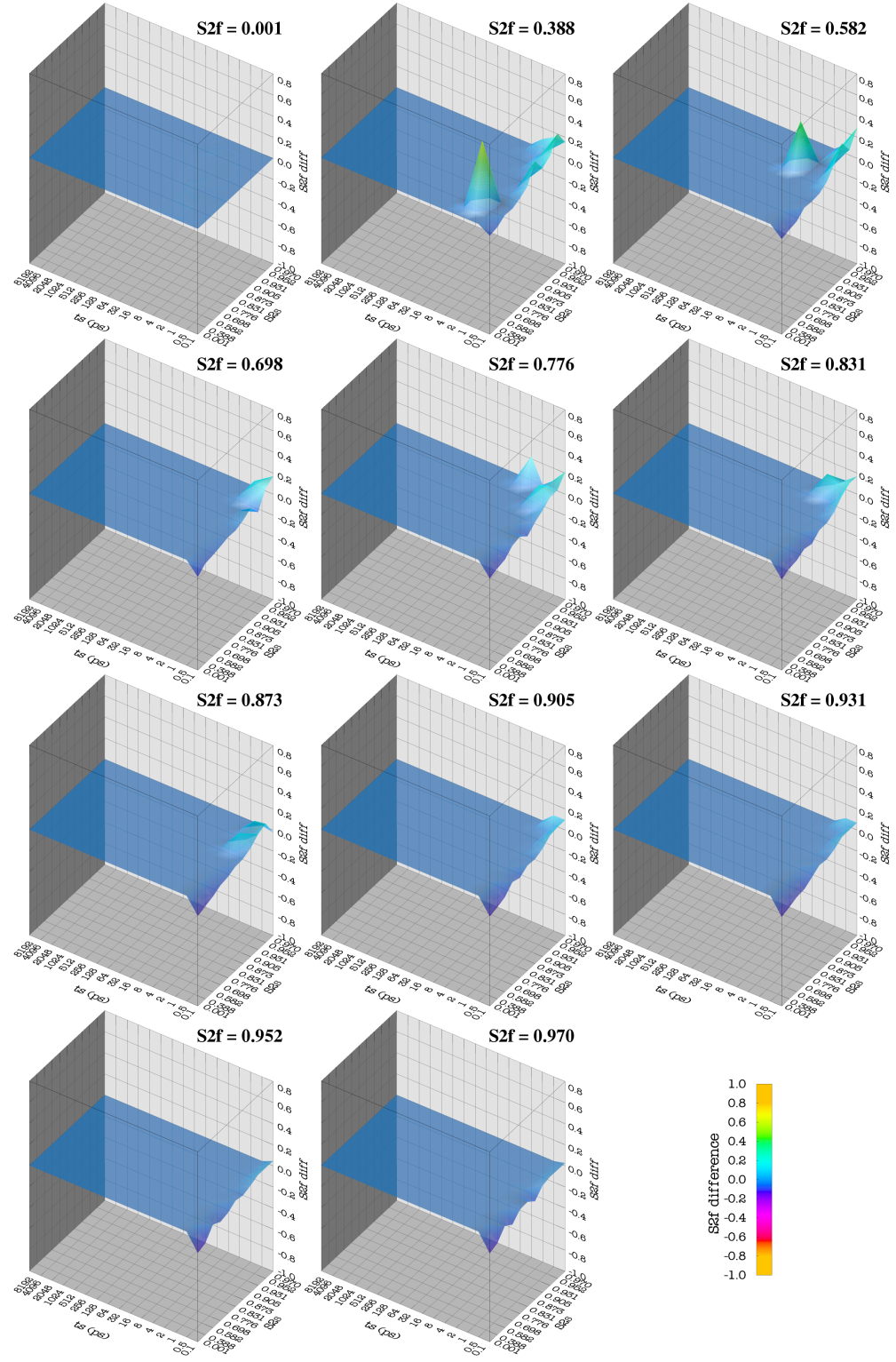


Figure S15: The results of the optimisation of the Double Motion Grid using relax presented as a reverse angle view of the  $S^2_f$  difference surface. For each grid point relax was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 0.005.



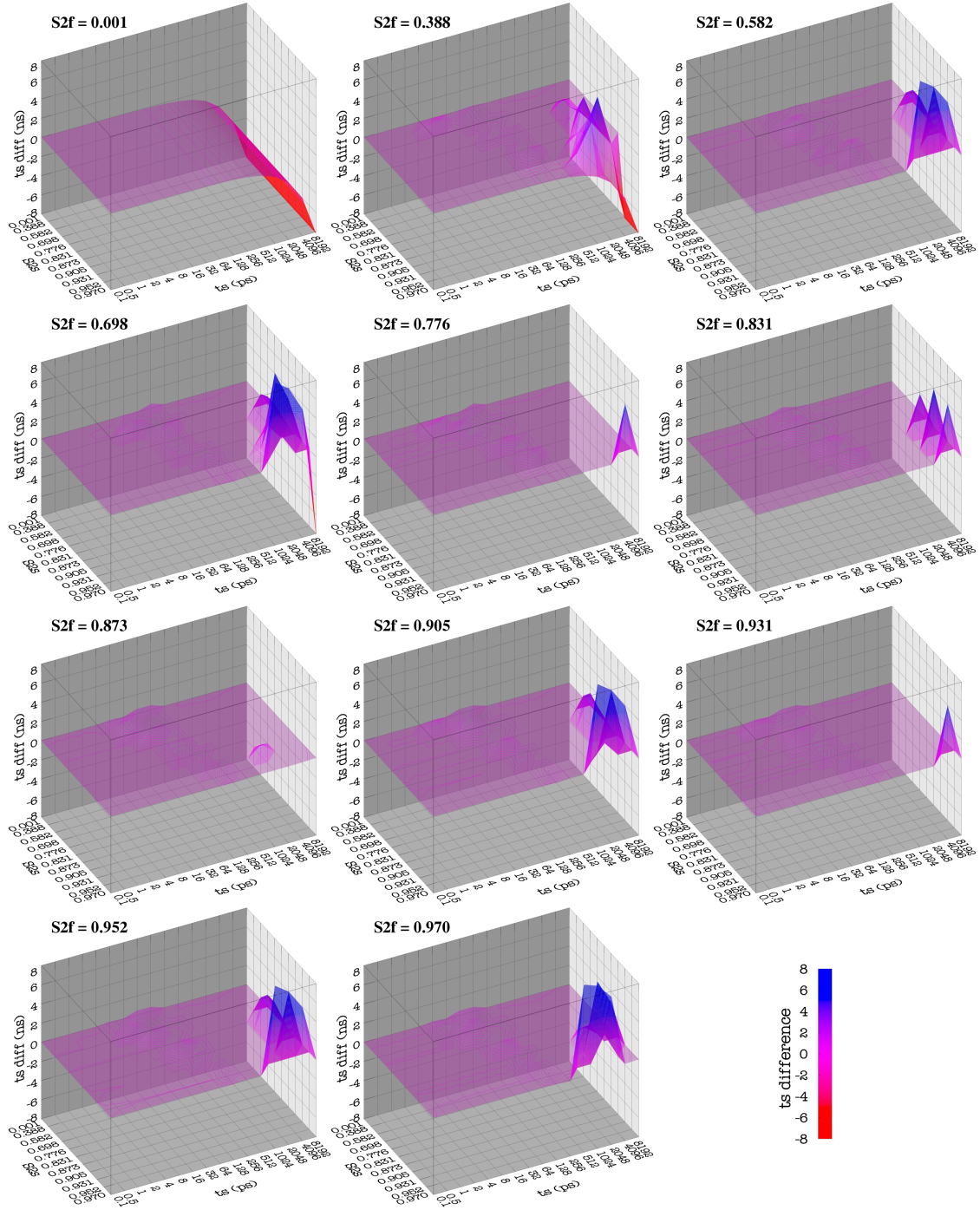


Figure S16: The results of the optimisation of the Double Motion Grid using Modelfree4 presented as a  $\tau_s$  difference surface. For each grid point Modelfree4 was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.

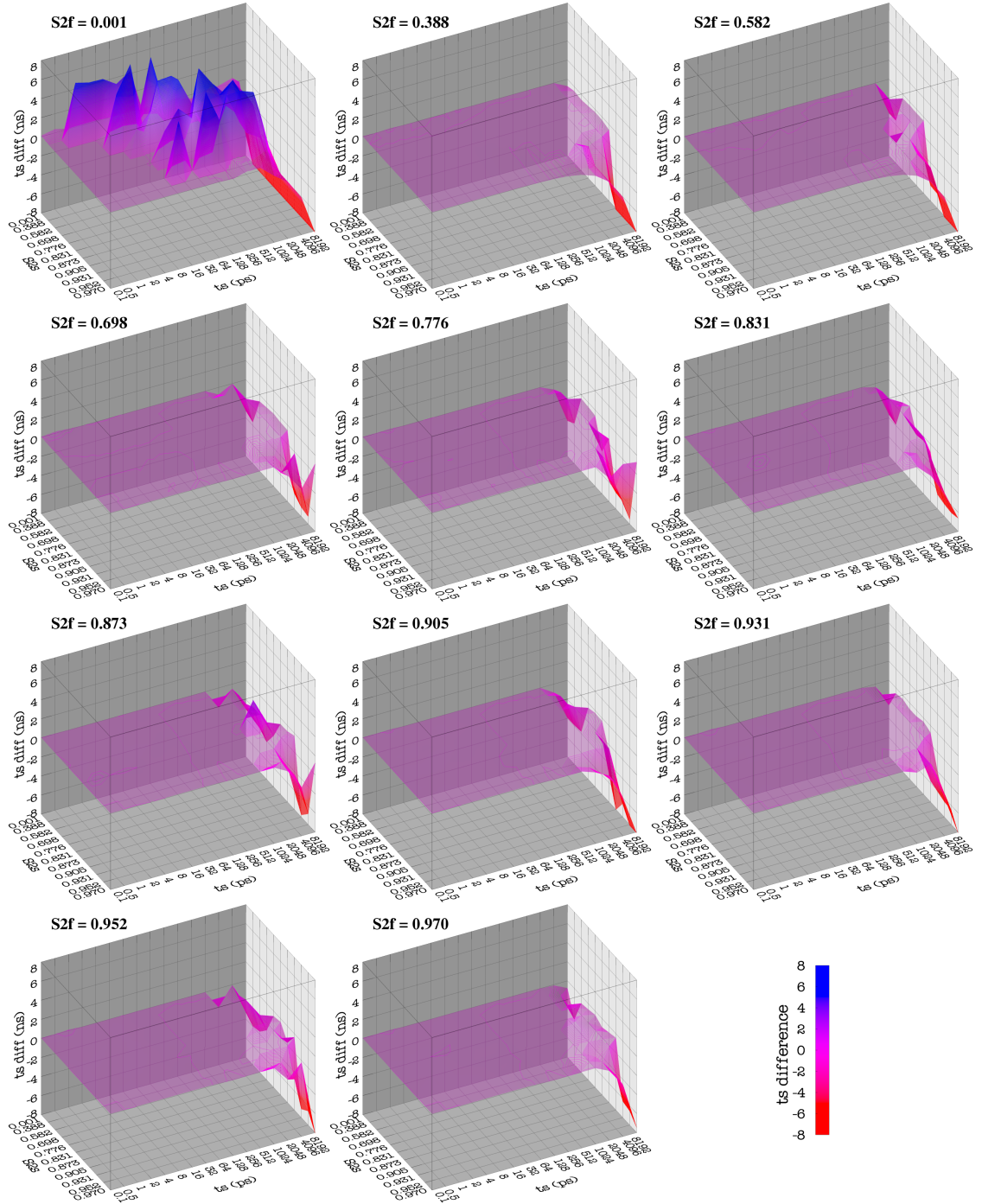


Figure S17: The results of the optimisation of the Double Motion Grid using Dasha presented as a  $\tau_s$  difference surface. For each grid point Dasha was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.

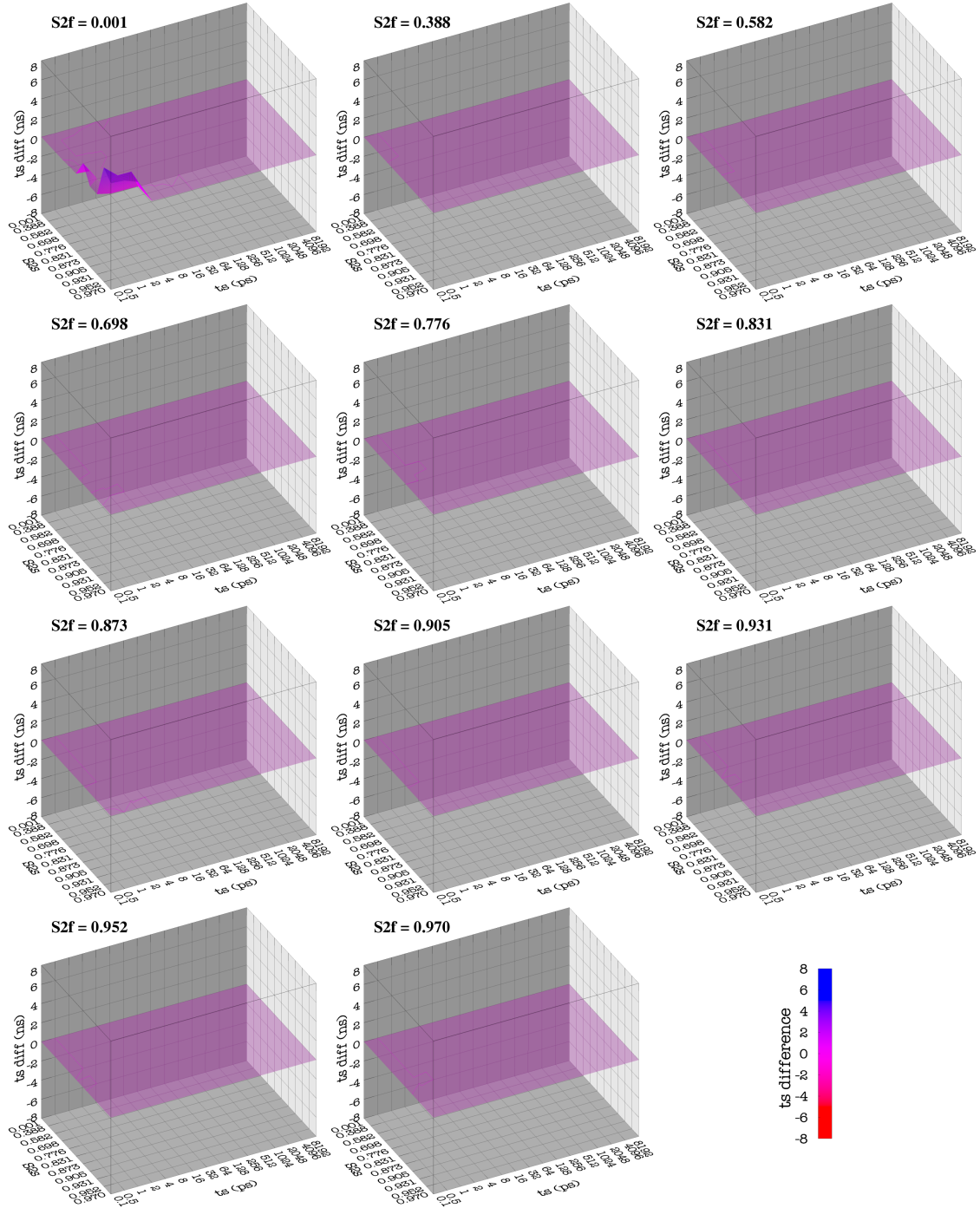


Figure S18: The results of the optimisation of the Double Motion Grid using relax presented as a  $\tau_s$  difference surface. For each grid point relax was used to optimise model  $m5$ . If the minimum has been found for each point the difference between the optimised and true parameter values should be zero. Positive and negative differences correspond to over and underestimation respectively. As a surface has been draped over the discrete differences perfect optimisation should result in a flat surface of zero height. The distance between the contour lines is 50 ps.

# Optimisation Theory

## The optimisation algorithms

Once a starting position for optimisation has been determined using a grid search the optimisation algorithm can be executed. The number of algorithms developed within the mathematical field of optimisation is considerable. They can nevertheless be grouped into one of a small number of major categories based on the fundamental principles of the technique. These include the line search methods, the trust region methods, and the conjugate gradient methods (Nocedal and Wright, 1999). The line search algorithms tested in this paper include: the steepest descent algorithm whereby search direction is simply the negative gradient; back-and-forth coordinate descent; the BFGS quasi-Newton algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970); the Newton search direction; and the Newton conjugate gradient method (Newton-CG). The Newton-Raphson-CG algorithm implemented within Dasha is similar to the Newton-CG technique as Newton optimisation is sometimes also known as the Newton-Raphson algorithm and, as documented in the source code, the Newton algorithm in Dasha is coupled to a conjugate gradient algorithm. The trust region algorithms tested include: the Cauchy point algorithm; the dogleg algorithm; Steihaug's modified conjugate gradient approach (Steihaug, 1983); and an exact trust region algorithm (Nocedal and Wright, 1999). Of the conjugate gradient techniques the following were tested: the Fletcher-Reeves algorithm which was the original conjugate gradient optimisation technique (Fletcher and Reeves, 1964); the Polak-Ribière method (Polak and Ribière, 1969); a modified Polak-Ribière method called the Polak-Ribière + method (Nocedal and Wright, 1999); and the Hestenes-Stiefel algorithm which originates from a formula in Hestenes and Stiefel (1952). Two other optimisation algorithms which cannot be classified within line search, trust region, or conjugate gradient categories were also investigated including the well known simplex optimisation algorithm and the commonly used Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963).

Once the search direction has been determined by the line search and conjugate gradient algorithms the minimum along that direction needs to be determined. Not to be confused with the methodology for determining the search direction, the line search itself is performed by an auxiliary step-length selection algorithm. Two of these techniques were investigated including the backtracking line search of Nocedal and Wright (1999) and the line search method of Moré and Thuente (1994).

A number of algorithms utilise the Hessian which is the matrix of second partial derivatives of the chi-squared equation. These techniques require the Hessian to be positive definite which may not always be the case as saddle points and other non-quadratic features of the space can be problematic. Two Hessian modification techniques which guarantee the matrix to be positive definite were investigated including an algorithm which utilises the Cholesky factorisation (Algorithm 6.3 of Nocedal and Wright (1999)) and the Gill, Murray, and Wright (GMW) algorithm (Gill et al., 1981).

To guarantee that the minimum will still be reached the implementation of constraints limiting the parameter values together with optimisation algorithms is not a triviality. For this to occur the space and its boundaries must remain smooth thereby allowing the algorithm to move along the boundary to either find the minimum along the limit or to slide along the limit and then move back into the centre of the constrained space once the curvature allows it. One of the most powerful approaches which was tested is the iterative Method of Multipliers (Nocedal and Wright, 1999), also known as the Augmented Lagrangian.

## Precision

To enable high precision optimisation the cutoffs used to terminate the minimisation algorithms were either set to very small values or turned off. Optimisation was only stopped after a chi-squared difference of  $1e^{-25}$  between two successive iterations. No step length or gradient difference cutoffs were implemented. The algorithms were also terminated if  $1e^7$  iterations were reached.



## Diagonal scaling

Model scaling can have a significant effect on the optimisation algorithm – a poorly scaled model can cause certain techniques to fail. When two parameters of the model lie on very different numeric scales the model is said to be poorly scaled. For example in model-free analysis the order of magnitude of the order parameters is one whereas for the internal correlation times the order of magnitude is between  $1e^{-12}$  to  $1e^{-8}$ . Most effected are the trust region algorithms – the multidimensional sphere of trust will either be completely ineffective against the correlation time parameters or severely restrict optimisation in the order parameter dimensions. In model-free analysis the significant scaling disparity sometimes causes optimisation to fail due to amplified effects of machine precision.

The model parameters can be scaled by supplying the optimisation algorithm with the scaled rather than unscaled parameters. When the chi-squared function, gradient, and Hessian are called the vector is then premultiplied with a diagonal matrix in which the diagonal elements are the scaling factors. For model-free analysis the scaling factor of one was used for the order parameter and a scaling factor of  $1e^{-12}$  was used for the correlation times. The  $R_{ex}$  parameter was scaled to be the chemical exchange rate of the first field strength.

## Comparison of numerous optimisation algorithms

### Testing the algorithms

The grid point chosen to represent the single motion spaces is  $S^2 = 0.831$ ,  $\tau_e = 256$  ps, and  $R_{ex} = 1.644 \text{ s}^{-1}$ . For simplicity this will be labelled the 256 ps RG point. Within the model-free space of this example, the curvature of which is illustrated in Figure S20, the gradient along the shallow valley is steep and hence most optimisation algorithms reach the minimum reasonably quickly (Table S1).

The ultimate test for the performance of optimisation algorithms within the model-free space is a model with motions on two timescales which are extremely close to each other. This can be provided by the grid point  $S^2 = 0.376$ ,  $S_f^2 = 0.970$ , and  $\tau_s = 4$  ps ( $S_s^2 = 0.388$  and  $\tau_f = 0$  ps) of the Double Motion Grid whereby the convolution of the  $S_f^2$  and  $\tau_s$  parameters creates a very long, curved, and deep tunnel through the space (Figure S21). This example will be labelled the 4 ps DMG point. The twisting of the tunnel is such that a decrease in the correlation time of the slower motion causes an increase in the order parameter of the faster meaning that optimisation within the model-free chi-squared space is a non-trivial problem. The 4 ps DMG point is utilised as a tool to contrast the performance and efficiency of different optimisation algorithms, it has no biological implications and model selection would never choose this model as it is statistically identical to model  $m2$ . The extremity of this example specifically amplifies the characteristics and differences between the optimisation techniques (Table S2) which occur to a lesser degree within the normal model-free parameter ranges (Table S1). The results of unconstrained minimisation are presented in Table S2 whereas the equivalent results from using the Augmented Lagrangian algorithm to constrain the parameters are shown in Table S3. This problem is similar to, yet much more complex than, the banana problem which is a popular challenging test of minimisation algorithms within the mathematical field of optimisation.

The results of the optimisation of two other grid points of the DMG are also presented here. The first, the 2 ps DMG point, is where  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 2$  ps. This example is an even more challenging optimisation problem than the 4 ps DMG point. The minimisation results are presented in Table S4 whereas the curvature of the model-free space is illustrated in Figure S22. The second point, labelled the 1024 ps DMG point, is where  $S^2 = 0.722$ ,  $S_f^2 = 0.931$ , and  $\tau_s = 1024$  ps ( $S_s^2 = 0.776$ ). The optimisation results are presented in Table S5. The separation of timescales is such that the space surrounding the minimum is very close to being quadratic and no curved tunnels are present (Figure S23).

## The conjugate gradient algorithms

When the major classes of optimisation algorithms are compared in Tables S1 and S2 with or without constraints the group of algorithms which perform the poorest, by either taking too many iterations or being incapable of finding the minimum, are the conjugate gradient algorithms. This is supported by the optimisation results of the grid points presented in the supplementary material. The conjugate gradient algorithms are techniques which are designed for large problems with many parameters. When challenged with the long, curved, and deep tunnels of the model-free space of the 4 ps DMG point, none of the techniques are successful in finding the minimum (Table S2). When unconstrained the Fletcher-Reeves, Polak-Ribière, and Polak-Ribière + algorithms all overshoot the minimum terminating at an  $S_f^2$  value close to 3 and a  $\tau_s$  value of about 1 ps. If the Augmented Lagrangian constraints are used together with the local optimisers (Table S3) they are prevented from overshooting yet they still do not reach the minimum. Despite the Hestenes-Stiefel algorithm not overshooting the mark this technique is also unsuccessful in reaching the minimum.

For the relatively simple space of the 256 ps RG point the backtracking line search performs very poorly (Table S1). This is probably because the ideal step length along the conjugate direction is beyond the initial step length yet the line search only finds a minimum inside the initial length. If the conjugate gradient techniques are to be used for model-free analysis the Moré and Thuente line search should be employed. However the CG techniques are outperformed by many of the line search and trust region algorithms within the model-free space.

## The trust region algorithms

For the Cauchy point algorithm the results for all examples match the theoretical predictions in that they are very similar to those of the steepest descent. As the maximal number of iterations has been attained in both Tables S1 and S2, preventing the minimum from being reached, as well as requiring one Hessian call per iteration of the algorithm, the technique performs the poorest of all algorithms tested.

Comparing the different matrices used for dogleg optimisation, the results of the unmodified Hessian for the 256 ps RG point are identical to those of the Cholesky and GWM Hessian modifications (Table S1). This is because the model-free space for this example is very close to being quadratic with the Hessian always positive definite. Thus the matrix needs never to be modified. These techniques, together with the more expensive BFGS method, all successfully find the minimum. However when the techniques are challenged with the curved, deep tunnel of the 4 ps DMG point, the optimisation results are very different. Only the GMW Hessian modification allows the minimum to be found. The unmodified Hessian should never be used as the curvature of this model-free space is not quadratic (Figure S21) causing the algorithms to not move far from their initial position. The Cholesky Hessian modification is both expensive and unreliable within this type of model-free space.

When optimising the 4 ps DMG point only three of the trust region methods are capable of reaching the minimum. These are the dogleg and exact trust region algorithms combined with the GMW Hessian modification as well as the CG-Steihaug method. Nevertheless the number of iterations, function calls, gradient calls, and Hessian calls are all greater than the Newton line search methods using the GMW Hessian modification. This is because the ellipsoid trust region limits the step length such that optimisation down the long, convoluted tunnel takes longer than the true Newton step. Therefore the trust region does not aid optimisation in the model-free space and only increases the computation time required.

## The line search methods

Because the rate of convergence of the steepest descent algorithm is only linear, theoretically, of all the line search methods it should perform the worst. Comparison of the results in both Tables S1 and S2 shows that this is indeed the case. Despite requiring close to 10 million iterations when unconstrained and reaching the maximum number of iterations imposed by the Method of Multipliers (156500 in this case) the local minimum is never reached in any of the examples. The back-and-forth coordinate descent (CD) method

performs better than the steepest descent as the minimum can be found for the 256 ps RG point. However CD is totally ineffective within the deep, convoluted tunnels of the double motion model-free spaces.

The results of the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) algorithm demonstrate the superlinear convergence of the quasi-Newton algorithms. Not only is the minimum found in both Tables S1 and S2 but, as no computationally expensive Hessian calls are made, the technique is relatively quick. Despite a decrease in the number of iterations the Newton-CG algorithm is computationally more expensive than the BFGS method. This is because both the number of function calls is doubled for the 4 ps DMG point and one Hessian call is made per iteration of the algorithm. Both techniques appear to have problems recovering from an overshoot when constraints are used (Table S3) with only the BFGS method using the Moré and Thuente line search finding the minimum in the constrained version of the 4 ps DMG point.

Of all the algorithms tested the Newton method in combination with the GMW Hessian modification exhibits the best performance. Not only is the minimum found in every example but the technique requires the least number of iterations to navigate the long, deep, curved tunnels of the double motion model-free spaces. Although the number of function calls is greater than the dogleg method with the GMW modification the number of Hessian calls in the double motion spaces is always less. Newton optimisation is therefore the best balance between computational intensity and reaching the minimum with the least number of iterations.

## The Hessian modifications

Within the deep, curved tunnels of the double motion model-free spaces the Cholesky Hessian modification fails to find good step directions. Although the number of iterations are much less than the steepest descent algorithm the technique performs relatively poorly requiring one to two orders of magnitude more iterations before the minimisation is terminated. In addition in almost all cases where the modification is used the minimum within the model-free space is never reached.

The GMW modified Cholesky factorisation performs extremely well allowing the minimum to be found in all example grid points when combined with the Newton, dogleg, or exact trust region optimisation algorithms. If the model-free Hessian is to be used in optimisation the matrix should be modified using the GMW algorithm to allow the minimum to be found.

## Step-length selection methods

The results of the two step-length selection algorithms vary depending upon which type of optimisation algorithm they are used with. For the conjugate gradient methods the backtracking line search is completely ineffective, hence the Moré and Thuente search should always be employed. On the other hand for the line search algorithms the backtracking algorithm is preferential. Not only are the number of iterations required to find the minimum slightly less but the number of gradient calls is halved. The inexact backtracking line search is much quicker than the exact Moré and Thuente search in finding the minimum for the line search algorithms.

## The simplex and Levenberg-Marquardt algorithms

Surprisingly, the simplex which wanders ameoboid-like through the model-free space is quite capable of sliding down the deep, curved tunnels of the double motion model-free spaces. The technique is able to find the minimum in all the examples tested. Although no gradient or Hessian calls are required, significantly decreasing the computation time per iteration, this is negated by the number of iterations and function calls required to reach the minimum being orders of magnitude greater than the Newton algorithm.

In the 256 ps RG point the Levenberg-Marquardt algorithm is extremely effective in finding the minimum in a small number of iterations. However for the 4 ps DMG point the algorithm completely fails due to the Levenberg-Marquardt matrix being singular and hence non-invertible. The algorithm is unable to move from the initial starting position found by the grid search. This catastrophic type of failure will be investigated below.

## Parameter deconvolution

Due to the construction of the extended model-free correlation function Clore et al. (1990) the use of the two model-free parameters  $S_f^2$  and  $S_s^2$  during optimisation results in a convolution of the space because of their relationship  $S_f^2 \cdot S_s^2 = S^2$ . Figure S19, which is a map of chi-squared space of the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 32$  ps of the DMG, illustrates this convolution and demonstrates a solution which simplifies the curvature of the model-free space. In the left two images the three dimensions of the model are  $\{S_f^2, S_s^2, \tau_s\}$ ; in the right two images the parameter  $S_s^2$  has been replaced by  $S^2$ . As optimisation slides down the curved tunnel of the  $\{S_f^2, S_s^2, \tau_s\}$  model the  $S_f^2$  value gradually increases while  $S_s^2$  decreases. The result is that by moving from the start of the tunnel to the minimum the values of  $S_f^2$  and  $S_s^2$  swap. On the other hand for the  $\{S^2, S_f^2, \tau_s\}$  model the  $S^2$  value is constant throughout the length of the tunnel. If high precision optimisation is used the final results are identical in both cases. The deconvolution by using the  $S^2$  rather than the  $S_s^2$  parameter results in the optimisation problem essentially collapsing from three to two dimensions. Minimisation is thereby simplified and the number of iterations required to find the minimum is decreased.



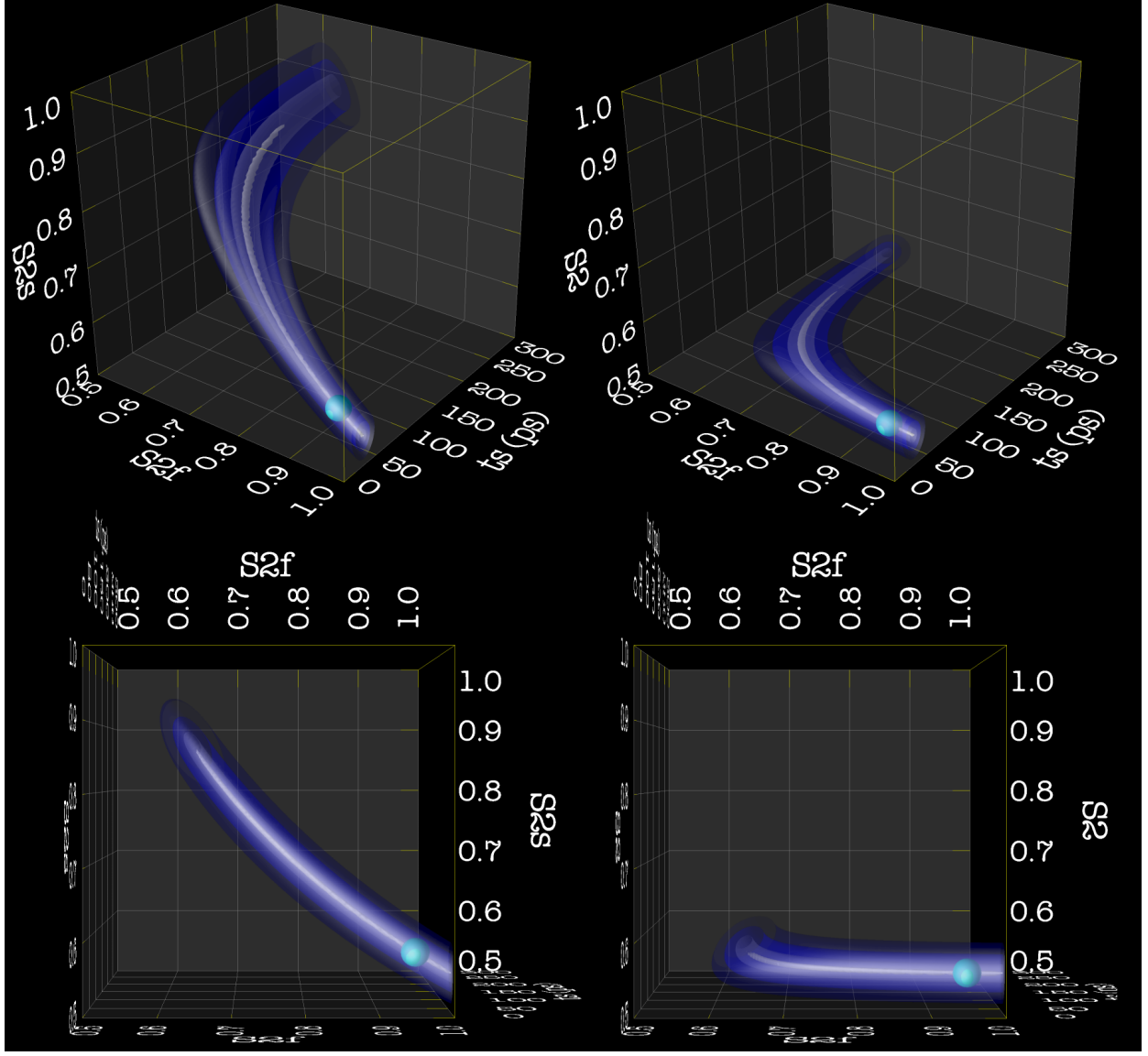


Figure S19: A demonstration of model-free parameter deconvolution. These maps of the chi-squared space correspond to the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 32$  ps of the Double Motion Grid. The cyan sphere indicates the position of the minimum within the deep, curved tunnels which traverse the model-free space. The curvature of the space is highlighted by four isosurfaces which correspond, from outermost to innermost, to chi-squared values of 50, 20, 5, and 0.5. In the model-free space the use of the two parameters  $S_f^2$  and  $S_s^2$  causes a convolution whereby, when heading down the tunnel, an increase in one parameter causes a decrease in the other. This is shown in the two images on the left. By using a model with the parameters  $\{S^2, S_f^2, \tau_s\}$  a deconvolution of the space occurs (as shown in the two images on the right). For this model the parameter  $S^2$  does not change when following the tunnel. This parameter deconvolution significantly simplifies optimisation of the model-free models.

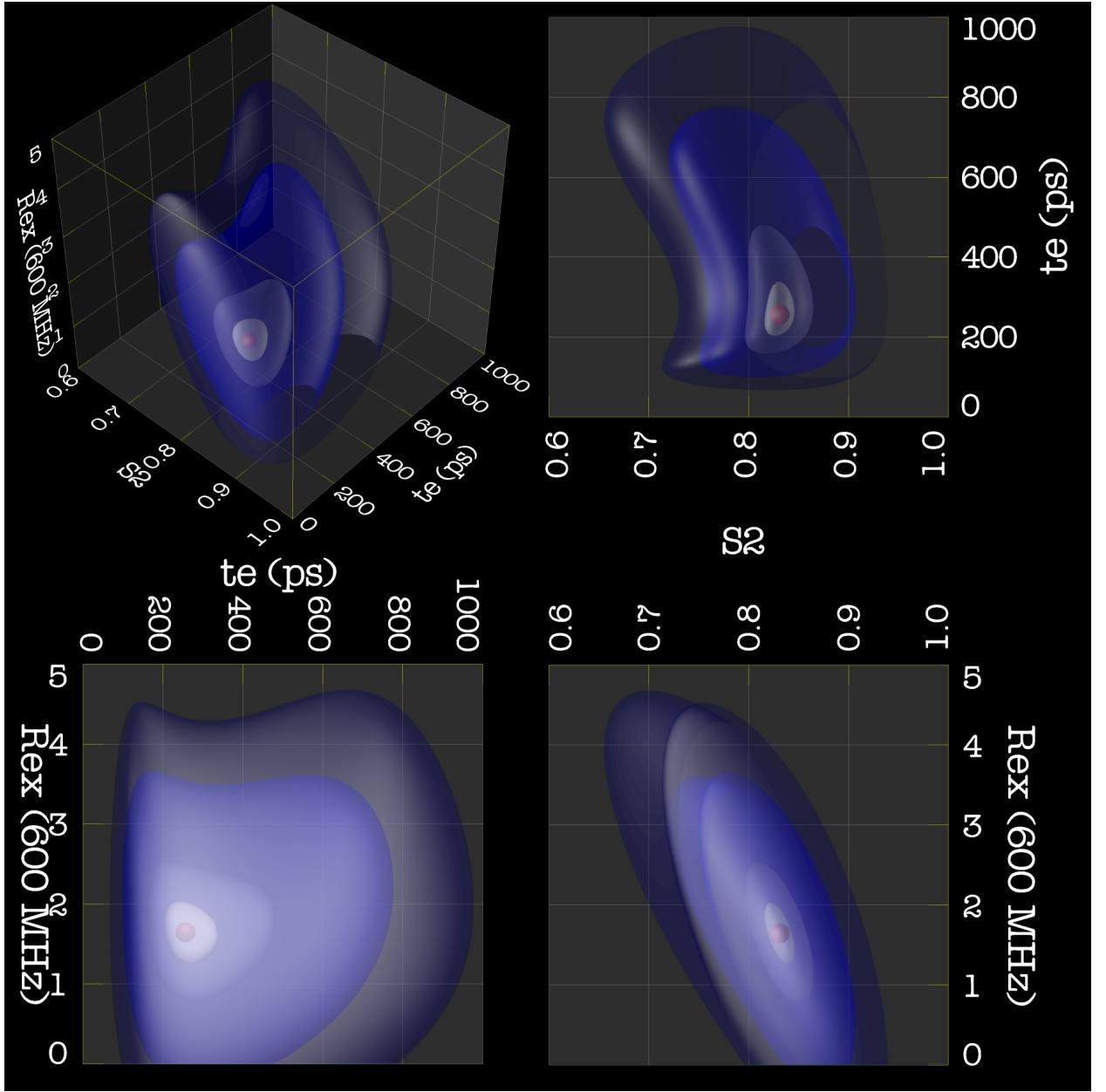


Figure S20: A map of the model-free space of the grid point  $S^2 = 0.831$ ,  $\tau_e = 256$  ps, and  $R_{ex} = 1.644$  s $^{-1}$ . The four isosurfaces of equal chi-squared values delineate the space where, from outermost to innermost, the chi-squared values are 100, 50, 10, and 2. The red sphere indicates the position of the true values.

Table S1: Comparison of the optimisation efficiency of various optimisation algorithms coupled to the Method of Multipliers constraint algorithm for the grid point  $S^2 = 0.831$ ,  $\tau_e = 256$  ps, and  $R_{ex} = 1.644 \text{ s}^{-1}$ .

| Algorithm                     | Auxiliary <sup>a</sup> | $S^2$ | $\tau_e$ (ps) | $R_{ex}$ | $i^b$  | $f^b$   | $g^b$  | $h^b$  | $\chi^2$             |
|-------------------------------|------------------------|-------|---------------|----------|--------|---------|--------|--------|----------------------|
| Line search algorithms        |                        |       |               |          |        |         |        |        |                      |
| Steepest descent              | Back                   | 0.826 | 201.6         | 1.752    | 156500 | 317584  | 156813 | 0      | 2.07                 |
| Steepest descent              | MT                     | 0.826 | 201.4         | 1.753    | 156500 | 313972  | 313972 | 0      | 2.09                 |
| Back-and-forth CD             | Back                   | 0.831 | 256.0         | 1.644    | 1323   | 3419    | 1330   | 0      | $1.75\text{e}^{-27}$ |
| Back-and-forth CD             | MT                     | 0.831 | 256.0         | 1.644    | 34     | 122     | 122    | 0      | $8.40\text{e}^{-7}$  |
| Quasi-Newton BFGS             | Back                   | 0.831 | 256.0         | 1.644    | 93     | 312     | 99     | 0      | $6.20\text{e}^{-27}$ |
| Quasi-Newton BFGS             | MT                     | 0.831 | 256.0         | 1.644    | 69     | 224     | 224    | 0      | $4.88\text{e}^{-28}$ |
| Newton CG                     | Back                   | 0.831 | 256.0         | 1.644    | 16     | 37      | 21     | 16     | $4.44\text{e}^{-28}$ |
| Newton CG                     | MT                     | 0.831 | 256.0         | 1.644    | 16     | 38      | 38     | 16     | $1.80\text{e}^{-28}$ |
| Newton                        | Back + Chol            | 0.831 | 256.0         | 1.644    | 6      | 16      | 10     | 6      | $2.46\text{e}^{-28}$ |
| Newton                        | Back + GMW             | 0.831 | 256.0         | 1.644    | 6      | 17      | 10     | 6      | $2.35\text{e}^{-28}$ |
| Newton                        | MT + Chol              | 0.831 | 256.0         | 1.644    | 6      | 16      | 16     | 6      | $2.46\text{e}^{-28}$ |
| Newton                        | MT + GMW               | 0.831 | 256.0         | 1.644    | 6      | 17      | 17     | 6      | $4.44\text{e}^{-28}$ |
| Trust region algorithms       |                        |       |               |          |        |         |        |        |                      |
| Cauchy point                  |                        | 0.826 | 202.2         | 1.750    | 156500 | 156813  | 156813 | 156500 | 2.02                 |
| Dogleg (BFGS)                 |                        | 0.831 | 256.0         | 1.644    | 173    | 180     | 180    | 0      | $4.13\text{e}^{-28}$ |
| Dogleg (Newton)               | Unmodified             | 0.831 | 256.0         | 1.644    | 6      | 10      | 10     | 6      | $6.63\text{e}^{-28}$ |
| Dogleg (Newton)               | Chol                   | 0.831 | 256.0         | 1.644    | 6      | 10      | 10     | 6      | $2.46\text{e}^{-28}$ |
| Dogleg (Newton)               | GMW                    | 0.831 | 256.0         | 1.644    | 6      | 10      | 10     | 6      | $6.63\text{e}^{-28}$ |
| CG-Steihaug                   |                        | 0.831 | 256.0         | 1.644    | 59     | 63      | 63     | 59     | $4.20\text{e}^{-28}$ |
| Exact trust region            | Unmodified             | 0.831 | 256.0         | 1.644    | 20     | 24      | 24     | 20     | $1.72\text{e}^{-28}$ |
| Exact trust region            | Chol                   | 0.831 | 256.0         | 1.644    | 20     | 24      | 24     | 20     | $1.72\text{e}^{-28}$ |
| Exact trust region            | GMW                    | 0.831 | 256.0         | 1.644    | 20     | 24      | 24     | 20     | $2.46\text{e}^{-28}$ |
| Conjugate gradient algorithms |                        |       |               |          |        |         |        |        |                      |
| Fletcher-Reeves               | Back                   | 0.830 | 238.5         | 1.668    | 156500 | 2390681 | 156813 | 0      | $1.59\text{e}^{-1}$  |
| Fletcher-Reeves               | MT                     | 0.831 | 256.0         | 1.644    | 54     | 286     | 286    | 0      | $6.72\text{e}^{-21}$ |
| Polak-Ribière                 | Back                   | 0.831 | 253.5         | 1.647    | 156437 | 2400832 | 156750 | 0      | $2.93\text{e}^{-3}$  |
| Polak-Ribière                 | MT                     | 0.831 | 256.0         | 1.644    | 360    | 1156    | 1156   | 0      | $2.24\text{e}^{-16}$ |
| Polak-Ribière +               | Back                   | 0.831 | 253.5         | 1.647    | 156437 | 2400832 | 156750 | 0      | $2.93\text{e}^{-3}$  |
| Polak-Ribière +               | MT                     | 0.831 | 256.0         | 1.644    | 360    | 1156    | 1156   | 0      | $2.24\text{e}^{-16}$ |
| Hestenes-Stiefel              | Back                   | 0.831 | 256.0         | 1.644    | 2779   | 39339   | 2786   | 0      | $2.01\text{e}^{-15}$ |
| Hestenes-Stiefel              | MT                     | 0.831 | 256.0         | 1.644    | 63     | 360     | 360    | 0      | $4.04\text{e}^{-21}$ |
| Miscellaneous algorithms      |                        |       |               |          |        |         |        |        |                      |
| Simplex                       |                        | 0.831 | 256.0         | 1.644    | 1635   | 2764    | 0      | 0      | $3.15\text{e}^{-28}$ |
| Levenberg-Marquardt           |                        | 0.831 | 256.0         | 1.644    | 8      | 14      | 14     | 0      | $4.47\text{e}^{-28}$ |

<sup>a</sup> The algorithms auxiliary to the main optimisation algorithm include: Back – backtracking line search; MT – Moré and Thuente line search; Unmodified – no Hessian modification; Chol – Cholesky Hessian modification with added multiple of the identity matrix; GMW – Gill, Murray, and Wright modified Cholesky Hessian modification algorithm.

<sup>b</sup>  $i$  is the number of iterations and  $f$ ,  $g$ , and  $h$  are the number of function, gradient, and Hessian calls respectively.

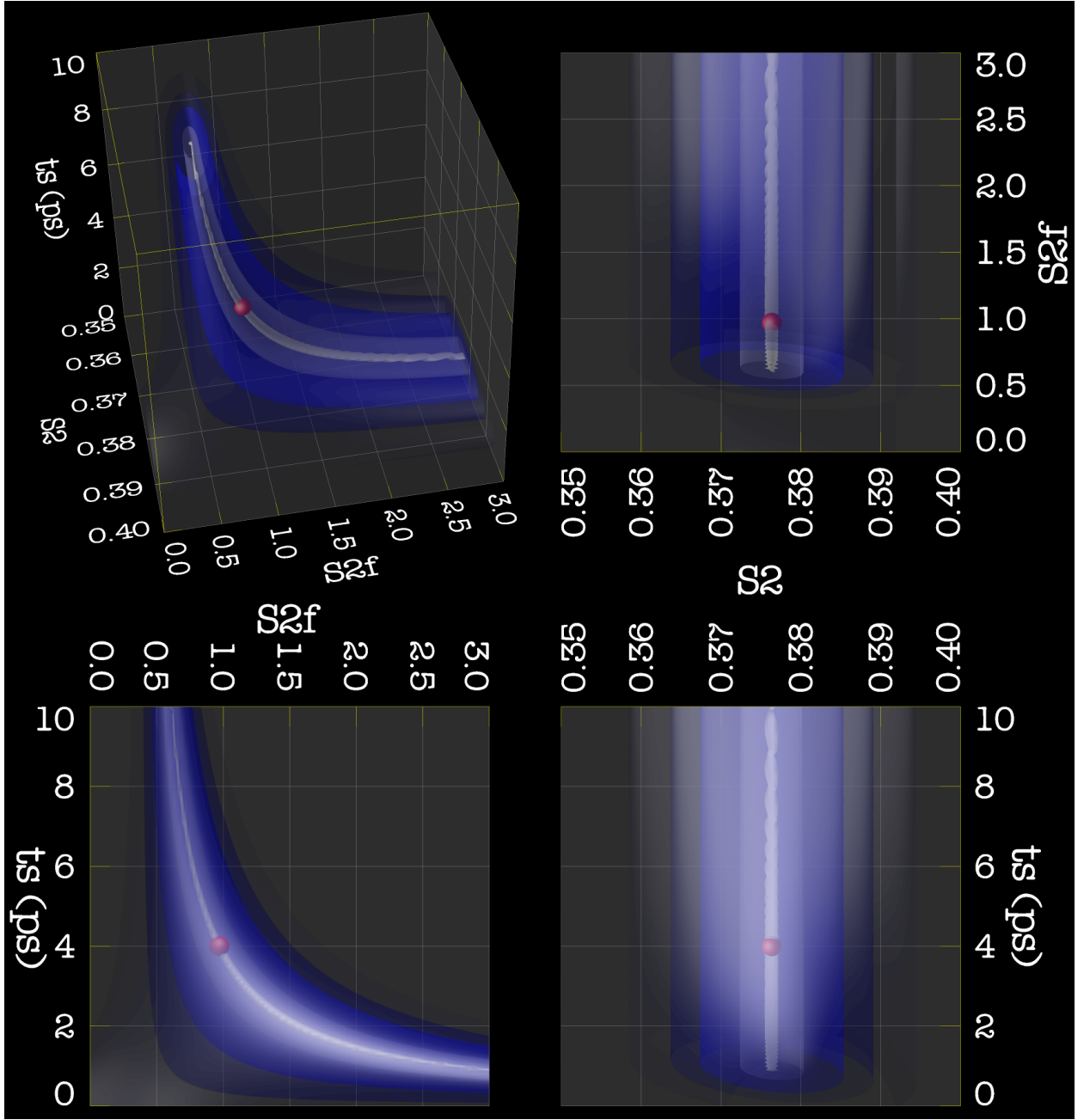


Figure S21: A map of the model-free space of the grid point  $S^2 = 0.376$ ,  $S_f^2 = 0.970$ , and  $\tau_s = 4$  ps ( $S_s^2 = 0.388$ ). The five isosurfaces of equal chi-squared values delineate the space where, from outermost to innermost, the chi-squared values are 20, 10, 5, 1, and 0.05. The red sphere indicates the position of the true values.

Table S2: Comparison of the optimisation efficiency of various unconstrained optimisation algorithms for the extremely difficult to minimise grid point  $S^2 = 0.376$ ,  $S_f^2 = 0.970$ , and  $\tau_s = 4$  ps ( $S_s^2 = 0.388$ ).

| Algorithm                        | Auxiliary <sup>a</sup> | $S^2$ | $S_f^2$ | $\tau_s$ (ps) | $i^b$             | $f^b$             | $g^b$             | $h^b$           | $\chi^2$             |
|----------------------------------|------------------------|-------|---------|---------------|-------------------|-------------------|-------------------|-----------------|----------------------|
| Line search algorithms           |                        |       |         |               |                   |                   |                   |                 |                      |
| Steepest descent                 | Back                   | 0.376 | 1.919   | 1.539         | 7.8e <sup>6</sup> | 1.6e <sup>7</sup> | 7.8e <sup>6</sup> | 0               | 5.47e <sup>-9</sup>  |
| Steepest descent                 | MT                     | 0.376 | 0.397   | 132.0         | 1e <sup>7</sup>   | 2e <sup>7</sup>   | 2e <sup>7</sup>   | 0               | 7.35e <sup>-3</sup>  |
| Back-and-forth CD                | Back                   | 0.400 | 1.400   | 0.000         | 1                 | 3                 | 2                 | 0               | 36.44                |
| Back-and-forth CD                | MT                     | 0.400 | 1.400   | 0.000         | 1                 | 3                 | 3                 | 0               | 36.44                |
| Quasi-Newton BFGS                | Back                   | 0.376 | 0.970   | 4             | 2292              | 5653              | 2293              | 0               | 1.64e <sup>-28</sup> |
| Quasi-Newton BFGS                | MT                     | 0.376 | 0.970   | 4             | 2273              | 5322              | 5322              | 0               | 3.50e <sup>-28</sup> |
| Newton CG                        | Back                   | 0.376 | 0.970   | 4             | 2040              | 10566             | 2041              | 2040            | 1.63e <sup>-28</sup> |
| Newton CG                        | MT                     | 0.376 | 0.970   | 4             | 2711              | 10366             | 10366             | 2711            | 2.61e <sup>-28</sup> |
| Newton                           | Back + Chol            | 0.376 | 1.919   | 1.539         | 89547             | 179133            | 89548             | 89547           | 5.47e <sup>-9</sup>  |
| Newton                           | Back + GMW             | 0.376 | 0.970   | 4             | 1327              | 3257              | 1328              | 1327            | 1.54e <sup>-28</sup> |
| Newton                           | MT + Chol              | 0.376 | 1.919   | 1.539         | 28265             | 76044             | 76044             | 28265           | 5.47e <sup>-9</sup>  |
| Newton                           | MT + GMW               | 0.376 | 0.970   | 4             | 1428              | 3326              | 3326              | 1428            | 1.63e <sup>-28</sup> |
| Trust region algorithms          |                        |       |         |               |                   |                   |                   |                 |                      |
| Cauchy point                     |                        | 0.376 | 1.765   | 1.710         | 1e <sup>7</sup>   | 1e <sup>7</sup>   | 1e <sup>7</sup>   | 1e <sup>7</sup> | 5.04e <sup>-9</sup>  |
| Dogleg (BFGS)                    |                        | 0.376 | 1.916   | 1.541         | 75674             | 75675             | 75675             | 0               | 5.47e <sup>-9</sup>  |
| Dogleg (Newton)                  | Unmodified             | 0.381 | 0.381   | 0.000         | 3                 | 4                 | 4                 | 3               | 11.96                |
| Dogleg (Newton)                  | Chol                   | 0.376 | 1.919   | 1.539         | 90891             | 90892             | 90892             | 90865           | 5.47e <sup>-9</sup>  |
| Dogleg (Newton)                  | GMW                    | 0.376 | 0.970   | 4             | 1710              | 1711              | 1711              | 1455            | 2.29e <sup>-28</sup> |
| CG-Steihaug                      |                        | 0.376 | 0.970   | 4             | 4454              | 4455              | 4455              | 3639            | 1.39e <sup>-22</sup> |
| Exact trust region <sup>c</sup>  | Unmodified             | 0.400 | 0.400   | 0.000         | 0                 | 1                 | 1                 | 1               | 36.44                |
| Exact trust region               | Chol                   | 0.376 | 1.919   | 1.539         | 84134             | 84135             | 84135             | 84104           | 5.47e <sup>-9</sup>  |
| Exact trust region               | GMW                    | 0.376 | 0.970   | 4             | 7737              | 7738              | 7738              | 7458            | 1.89e <sup>-28</sup> |
| Conjugate gradient algorithms    |                        |       |         |               |                   |                   |                   |                 |                      |
| Fletcher-Reeves                  | Back                   | 0.376 | 2.653   | 1.042         | 314               | 5474              | 315               | 0               | 6.55e <sup>-9</sup>  |
| Fletcher-Reeves                  | MT                     | 0.376 | 2.827   | 0.968         | 19                | 200               | 200               | 0               | 6.68e <sup>-9</sup>  |
| Polak-Ribière                    | Back                   | 0.382 | 2.724   | 1.196         | 29                | 545               | 30                | 0               | 3.03                 |
| Polak-Ribière                    | MT                     | 0.376 | 3.000   | 0.905         | 22                | 209               | 209               | 0               | 6.79e <sup>-9</sup>  |
| Polak-Ribière +                  | Back                   | 0.382 | 2.724   | 1.196         | 29                | 545               | 30                | 0               | 3.03                 |
| Polak-Ribière +                  | MT                     | 0.376 | 3.000   | 0.905         | 22                | 209               | 209               | 0               | 6.79e <sup>-9</sup>  |
| Hestenes-Stiefel                 | Back                   | 0.381 | 0.400   | 0.202         | 16                | 314               | 17                | 0               | 11.92                |
| Hestenes-Stiefel                 | MT                     | 0.376 | 0.515   | 17.22         | 820               | 3498              | 3498              | 0               | 2.30e <sup>-6</sup>  |
| Miscellaneous algorithms         |                        |       |         |               |                   |                   |                   |                 |                      |
| Simplex                          |                        | 0.376 | 0.970   | 4             | 5825              | 9961              | 0                 | 0               | 4.88e <sup>-26</sup> |
| Levenberg-Marquardt <sup>c</sup> |                        | 0.400 | 0.400   | 0.000         | 0                 | 1                 | 1                 | 1               | 36.44                |

<sup>a</sup> The algorithms auxiliary to the main optimisation algorithm include: Back – backtracking line search; MT – Moré and Thuente line search; Unmodified – no Hessian modification; Chol – Cholesky Hessian modification with added multiple of the identity matrix; GMW – Gill, Murray, and Wright modified Cholesky Hessian modification algorithm.

<sup>b</sup>  $i$  is the number of iterations and  $f$ ,  $g$ , and  $h$  are the number of function, gradient, and Hessian calls respectively.

<sup>c</sup> Algorithm failure due to the Hessian not being positive definite or the Levenberg-Marquardt matrix being singular and hence non-invertible.



Table S3: Comparison of the optimisation efficiency of various optimisation algorithms coupled to the Method of Multipliers constraint algorithm for the grid point  $S^2 = 0.376$ ,  $S_f^2 = 0.970$ , and  $\tau_s = 4$  ps ( $S_s^2 = 0.388$ ).

| Algorithm                        | Auxiliary <sup>a</sup> | $S^2$ | $S_f^2$ | $\tau_s$ (ps) | $i^b$ | $f^b$ | $g^b$ | $h^b$ | $\chi^2$             |
|----------------------------------|------------------------|-------|---------|---------------|-------|-------|-------|-------|----------------------|
| Line search algorithms           |                        |       |         |               |       |       |       |       |                      |
| Steepest descent                 | Back                   | 0.380 | 1.000   | 0.862         | 19885 | 41231 | 19928 | 0     | 7.14                 |
| Steepest descent                 | MT                     | 0.380 | 1.000   | 0.620         | 18155 | 37800 | 37800 | 0     | 8.37                 |
| Back-and-forth CD                | Back                   | 0.376 | 0.995   | 3.840         | 82    | 983   | 91    | 0     | 4.64e <sup>-11</sup> |
| Back-and-forth CD                | MT                     | 0.376 | 1.000   | 3.807         | 87    | 625   | 625   | 0     | 6.65e <sup>-11</sup> |
| Quasi-Newton BFGS                | Back                   | 0.376 | 1.000   | 3.807         | 61    | 343   | 68    | 0     | 6.64e <sup>-11</sup> |
| Quasi-Newton BFGS                | MT                     | 0.376 | 0.970   | 4             | 432   | 1150  | 1150  | 0     | 2.15e <sup>-26</sup> |
| Newton CG                        | Back                   | 0.376 | 1.000   | 3.807         | 11515 | 23231 | 11540 | 11515 | 6.65e <sup>-11</sup> |
| Newton CG                        | MT                     | 0.376 | 1.000   | 3.807         | 11504 | 23159 | 23159 | 11504 | 6.65e <sup>-11</sup> |
| Newton                           | Back + Chol            | 0.376 | 1.000   | 3.807         | 12899 | 25960 | 12926 | 12899 | 6.65e <sup>-11</sup> |
| Newton                           | Back + GMW             | 0.376 | 0.970   | 4             | 226   | 635   | 232   | 226   | 2.48e <sup>-28</sup> |
| Newton                           | MT + Chol              | 0.376 | 0.970   | 4             | 4482  | 12909 | 12909 | 4482  | 1.41e <sup>-28</sup> |
| Newton                           | MT + GMW               | 0.376 | 0.970   | 4             | 238   | 606   | 606   | 238   | 1.89e <sup>-28</sup> |
| Trust region algorithms          |                        |       |         |               |       |       |       |       |                      |
| Cauchy point                     |                        | 0.379 | 1.000   | 1.328         | 21808 | 21855 | 21855 | 21808 | 5.06                 |
| Dogleg (BFGS)                    |                        | 0.376 | 1.000   | 3.807         | 189   | 195   | 195   | 0     | 6.63e <sup>-11</sup> |
| Dogleg (Newton)                  | Unmodified             | 0.381 | 0.531   | 0.150         | 3446  | 3454  | 3454  | 2193  | 11.73                |
| Dogleg (Newton)                  | Chol                   | 0.380 | 1.000   | 0.619         | 12957 | 12984 | 12984 | 12885 | 8.38                 |
| Dogleg (Newton)                  | GMW                    | 0.376 | 0.970   | 4             | 345   | 351   | 351   | 255   | 1.94e <sup>-28</sup> |
| CG-Steihaug                      |                        | 0.376 | 0.970   | 4             | 1281  | 1287  | 1287  | 902   | 4.11e <sup>-19</sup> |
| Exact trust region <sup>c</sup>  | Unmodified             | 0.381 | 0.481   | 0.100         | 2     | 4     | 4     | 4     | 11.86                |
| Exact trust region               | Chol                   | 0.376 | 1.000   | 3.807         | 13041 | 13069 | 13069 | 12934 | 6.65e <sup>-11</sup> |
| Exact trust region               | GMW                    | 0.376 | 0.970   | 4             | 329   | 335   | 335   | 255   | 1.89e <sup>-28</sup> |
| Conjugate gradient algorithms    |                        |       |         |               |       |       |       |       |                      |
| Fletcher-Reeves                  | Back                   | 0.376 | 0.974   | 3.972         | 335   | 5865  | 341   | 0     | 1.46e <sup>-12</sup> |
| Fletcher-Reeves                  | MT                     | 0.376 | 1.000   | 3.810         | 38    | 468   | 468   | 0     | 6.42e <sup>-11</sup> |
| Polak-Ribière                    | Back                   | 0.376 | 0.633   | 9.247         | 455   | 8082  | 460   | 0     | 1.42e <sup>-7</sup>  |
| Polak-Ribière                    | MT                     | 0.376 | 0.999   | 3.812         | 55    | 475   | 475   | 0     | 6.36e <sup>-11</sup> |
| Polak-Ribière +                  | Back                   | 0.376 | 0.633   | 9.247         | 455   | 8082  | 460   | 0     | 1.42e <sup>-7</sup>  |
| Polak-Ribière +                  | MT                     | 0.376 | 0.999   | 3.812         | 55    | 475   | 475   | 0     | 6.36e <sup>-11</sup> |
| Hestenes-Stiefel                 | Back                   | 0.376 | 0.875   | 4.760         | 282   | 4831  | 287   | 0     | 1.30e <sup>-9</sup>  |
| Hestenes-Stiefel                 | MT                     | 0.376 | 0.993   | 3.853         | 32    | 217   | 217   | 0     | 3.91e <sup>-11</sup> |
| Miscellaneous algorithms         |                        |       |         |               |       |       |       |       |                      |
| Simplex                          |                        | 0.376 | 0.970   | 4             | 4401  | 7656  | 0     | 0     | 2.09e <sup>-28</sup> |
| Levenberg-Marquardt <sup>c</sup> |                        | 0.400 | 0.400   | 0.000         | 0     | 1     | 1     | 1     | 36.44                |

<sup>a</sup> The algorithms auxiliary to the main optimisation algorithm include: Back – backtracking line search; MT – More and Thuente line search; Unmodified – no Hessian modification; Chol – Cholesky Hessian modification with added multiple of the identity matrix; GMW – Gill, Murray, and Wright modified Cholesky Hessian modification algorithm.

<sup>b</sup>  $i$  is the number of iterations while  $f$ ,  $g$ , and  $h$  are the number of function, gradient, and Hessian calls respectively.

<sup>c</sup> Algorithm failure due to the Hessian not being positive definite or the Levenberg-Marquardt matrix being singular and hence non-invertible.

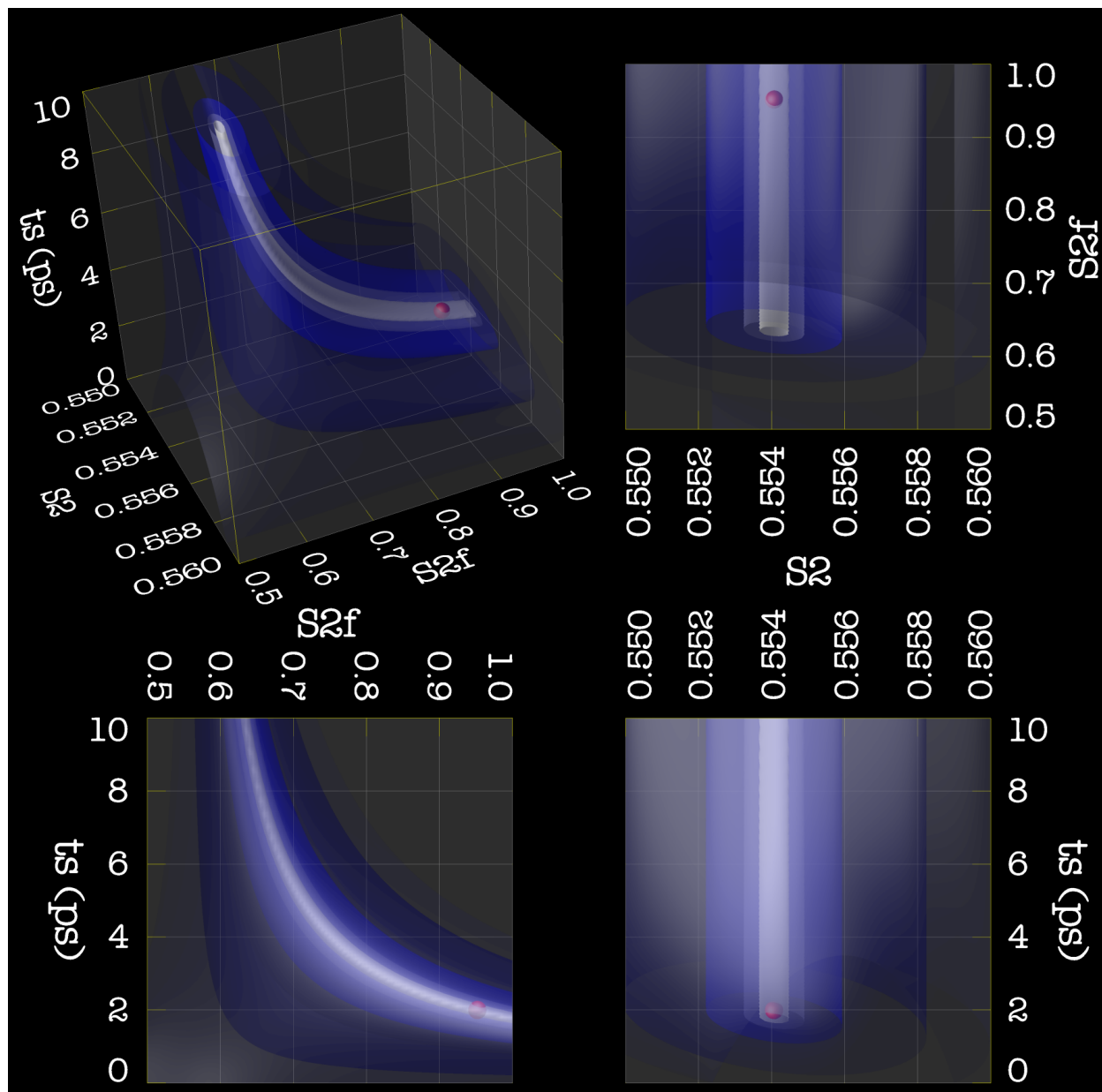


Figure S22: A map of the model-free space of the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 2$  ps. The five isosurfaces of equal chi-squared values delineate the space where, from outermost to innermost, the chi-squared values are 1, 0.5, 0.1, 0.02, and 0.005. The red sphere indicates the position of the true values.

Table S4: Comparison of the optimisation efficiency of various constrained optimisation algorithms for the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 2$  ps.

| Algorithm                        | Auxiliary <sup>a</sup> | $S_f^2$ | $S_s^2$ | $\tau_s$ (ps) | $i^b$ | $f^b$  | $g^b$ | $h^b$ | $\chi^2$             |
|----------------------------------|------------------------|---------|---------|---------------|-------|--------|-------|-------|----------------------|
| Line search algorithms           |                        |         |         |               |       |        |       |       |                      |
| Steepest descent                 | Back                   | 1.000   | 0.555   | 0.392         | 16346 | 33908  | 16385 | 0     | $3.92\text{e}^{-1}$  |
| Steepest descent                 | MT                     | 1.000   | 0.554   | 1.415         | 15041 | 32170  | 32170 | 0     | $2.76\text{e}^{-2}$  |
| Back-and-forth CD                | Back                   | 0.820   | 0.676   | 2.999         | 76    | 647    | 82    | 0     | $3.97\text{e}^{-11}$ |
| Back-and-forth CD                | MT                     | 1.000   | 0.554   | 1.785         | 117   | 876    | 876   | 0     | $1.06\text{e}^{-12}$ |
| Quasi-Newton BFGS                | Back                   | 0.998   | 0.555   | 1.793         | 59    | 364    | 67    | 0     | $9.79\text{e}^{-13}$ |
| Quasi-Newton BFGS                | MT                     | 1.000   | 0.554   | 1.785         | 40    | 198    | 198   | 0     | $1.06\text{e}^{-12}$ |
| Newton CG                        | Back                   | 1.000   | 0.554   | 1.785         | 28951 | 58529  | 29011 | 28951 | $1.06\text{e}^{-12}$ |
| Newton CG                        | MT                     | 1.000   | 0.555   | 0.445         | 28925 | 57919  | 57919 | 28925 | $3.63\text{e}^{-1}$  |
| Newton                           | Back + Chol            | 1.000   | 0.554   | 1.785         | 32508 | 65289  | 32576 | 32508 | $1.06\text{e}^{-12}$ |
| Newton                           | Back + GMW             | 0.952   | 0.582   | 2             | 976   | 2511   | 983   | 976   | $1.22\text{e}^{-28}$ |
| Newton                           | MT + Chol              | 1.000   | 0.554   | 1.785         | 10668 | 28755  | 28755 | 10668 | $1.06\text{e}^{-12}$ |
| Newton                           | MT + GMW               | 0.952   | 0.582   | 2             | 1041  | 2586   | 2586  | 1041  | $1.41\text{e}^{-28}$ |
| Trust region algorithms          |                        |         |         |               |       |        |       |       |                      |
| Cauchy point                     |                        | 1.000   | 0.555   | 0.445         | 27792 | 27849  | 27849 | 27792 | $3.62\text{e}^{-1}$  |
| Dogleg (BFGS)                    |                        | 1.000   | 0.554   | 1.785         | 167   | 175    | 175   | 0     | $1.05\text{e}^{-12}$ |
| Dogleg (Newton)                  | Unmodified             | 0.662   | 0.839   | 0.106         | 1788  | 1793   | 1793  | 1118  | $6.25\text{e}^{-1}$  |
| Dogleg (Newton)                  | Chol                   | 1.000   | 0.554   | 1.785         | 47054 | 47152  | 47152 | 46907 | $1.06\text{e}^{-12}$ |
| Dogleg (Newton)                  | GMW                    | 0.952   | 0.582   | 2             | 1371  | 1378   | 1378  | 1079  | $1.25\text{e}^{-28}$ |
| CG-Steihaug                      |                        | 0.952   | 0.582   | 1.999         | 2059  | 2067   | 2067  | 1582  | $4.01\text{e}^{-18}$ |
| Exact trust region <sup>c</sup>  | Unmodified             | 0.656   | 0.847   | 0.100         | 3     | 5      | 5     | 5     | $6.27\text{e}^{-1}$  |
| Exact trust region               | Chol                   | 0.952   | 0.582   | 2             | 33906 | 33976  | 33976 | 33599 | $1.22\text{e}^{-28}$ |
| Exact trust region               | GMW                    | 0.952   | 0.582   | 2             | 1445  | 1452   | 1452  | 1124  | $1.32\text{e}^{-28}$ |
| Conjugate gradient algorithms    |                        |         |         |               |       |        |       |       |                      |
| Fletcher-Reeves                  | Back                   | 0.988   | 0.561   | 1.833         | 5962  | 100653 | 5975  | 0     | $6.50\text{e}^{-13}$ |
| Fletcher-Reeves                  | MT                     | 1.000   | 0.554   | 1.785         | 20    | 297    | 297   | 0     | $1.06\text{e}^{-12}$ |
| Polak-Ribière                    | Back                   | 0.933   | 0.594   | 2.102         | 5167  | 98951  | 5179  | 0     | $2.79\text{e}^{-13}$ |
| Polak-Ribière                    | MT                     | 1.000   | 0.554   | 1.785         | 18    | 188    | 188   | 0     | $1.06\text{e}^{-12}$ |
| Polak-Ribière +                  | Back                   | 0.933   | 0.594   | 2.102         | 5167  | 98951  | 5179  | 0     | $2.79\text{e}^{-13}$ |
| Polak-Ribière +                  | MT                     | 1.000   | 0.554   | 1.785         | 18    | 188    | 188   | 0     | $1.06\text{e}^{-12}$ |
| Hestenes-Stiefel                 | Back                   | 0.942   | 0.588   | 2.049         | 523   | 8823   | 529   | 0     | $6.34\text{e}^{-14}$ |
| Hestenes-Stiefel                 | MT                     | 1.000   | 0.554   | 1.785         | 23    | 179    | 179   | 0     | $1.06\text{e}^{-12}$ |
| Miscellaneous algorithms         |                        |         |         |               |       |        |       |       |                      |
| Simplex                          |                        | 0.952   | 0.582   | 2             | 18849 | 32936  | 0     | 0     | $1.02\text{e}^{-26}$ |
| Levenberg-Marquardt <sup>c</sup> |                        | 0.600   | 1.000   | 0.000         | 290   | 291    | 150   | 0     | 64.22                |

<sup>a</sup> The algorithms auxiliary to the main optimisation algorithm include: Back – backtracking line search; MT – Moré and Thuente line search; Unmodified – no Hessian modification; Chol – Cholesky Hessian modification with added multiple of the identity matrix; GMW – Gill, Murray, and Wright modified Cholesky Hessian modification algorithm.

<sup>b</sup>  $i$  is the number of iterations and  $f$ ,  $g$ , and  $h$  are the number of function, gradient, and Hessian calls respectively.

<sup>c</sup> Algorithm failure due to the Hessian not being positive definite or the Levenberg-Marquardt matrix being singular and hence non-invertible.

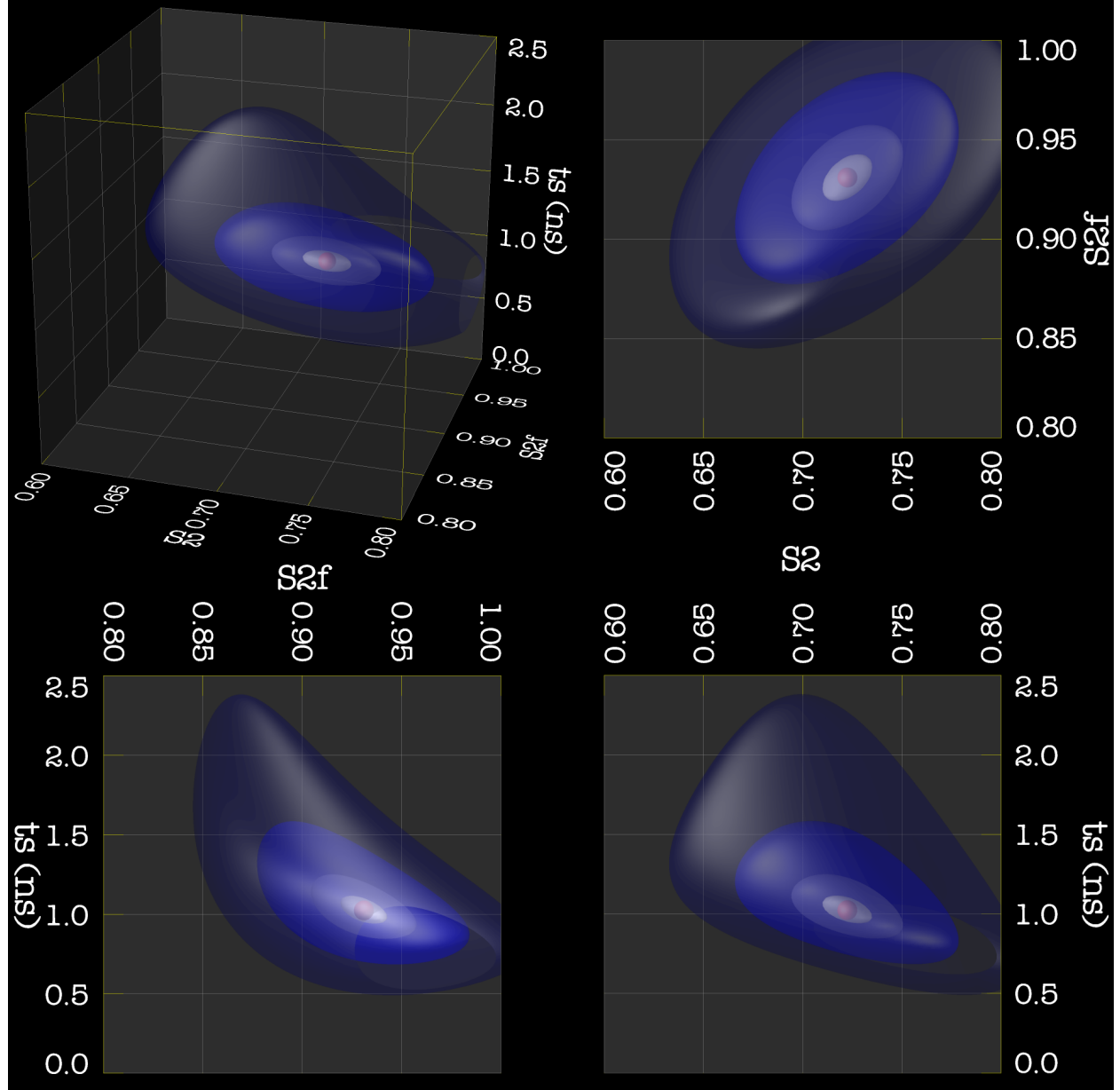


Figure S23: A map of the model-free space of the grid point  $S^2 = 0.722$ ,  $S_f^2 = 0.931$ , and  $\tau_s = 1024$  ps ( $S_s^2 = 0.776$ ). The four isosurfaces of equal chi-squared values delineate the space where, from outermost to innermost, the chi-squared values are 50, 20, 5, and 1. The red sphere indicates the position of the true values.

Table S5: Comparison of the optimisation efficiency of various constrained optimisation algorithms for the grid point  $S^2 = 0.722$ ,  $S_f^2 = 0.931$ , and  $\tau_s = 1024$  ps ( $S_s^2 = 0.776$ ).

| Algorithm                       | Auxiliary <sup>a</sup> | $S^2$ | $S_f^2$ | $\tau_s$ (ps) | $i^b$  | $f^b$  | $g^b$  | $h^b$  | $\chi^2$             |
|---------------------------------|------------------------|-------|---------|---------------|--------|--------|--------|--------|----------------------|
| Line search algorithms          |                        |       |         |               |        |        |        |        |                      |
| Steepest descent                | Back                   | 0.780 | 0.945   | 502.3         | 156500 | 317619 | 156813 | 0      | 47.88                |
| Steepest descent                | MT                     | 0.780 | 0.945   | 501.9         | 156500 | 313952 | 313952 | 0      | 47.96                |
| Back-and-forth CD               | Back                   | 0.722 | 0.931   | 1024.0        | 3153   | 7232   | 3163   | 0      | 3.14e <sup>-26</sup> |
| Back-and-forth CD               | MT                     | 0.722 | 0.931   | 1024.0        | 170    | 628    | 628    | 0      | 1.03e <sup>-23</sup> |
| Quasi-Newton BFGS               | Back                   | 0.722 | 0.931   | 1024.0        | 77     | 278    | 83     | 0      | 5.71e <sup>-17</sup> |
| Quasi-Newton BFGS               | MT                     | 0.722 | 0.931   | 1024.0        | 84     | 328    | 328    | 0      | 5.38e <sup>-27</sup> |
| Newton CG                       | Back                   | 0.722 | 0.931   | 1024.0        | 35     | 77     | 40     | 35     | 1.57e <sup>-28</sup> |
| Newton CG                       | MT                     | 0.722 | 0.931   | 1024.0        | 33     | 73     | 73     | 33     | 1.78e <sup>-28</sup> |
| Newton                          | Back + Chol            | 0.699 | 0.869   | 2400.0        | 156500 | 313313 | 156813 | 156500 | 50.48                |
| Newton                          | Back + GMW             | 0.722 | 0.931   | 1024.0        | 13     | 43     | 17     | 13     | 1.43e <sup>-28</sup> |
| Newton                          | MT + Chol              | 0.699 | 0.869   | 2399.9        | 156500 | 420590 | 420590 | 156500 | 50.47                |
| Newton                          | MT + GMW               | 0.722 | 0.931   | 1024.0        | 14     | 64     | 64     | 14     | 1.46e <sup>-28</sup> |
| Trust region algorithms         |                        |       |         |               |        |        |        |        |                      |
| Cauchy point                    |                        | 0.780 | 0.945   | 503.0         | 156500 | 156813 | 156813 | 156500 | 47.76                |
| Dogleg (BFGS)                   |                        | 0.722 | 0.931   | 1024.0        | 172    | 179    | 179    | 0      | 9.81e <sup>-21</sup> |
| Dogleg (Newton)                 | Unmodified             | 0.698 | 0.865   | 2622.3        | 209    | 212    | 212    | 83     | 56.05                |
| Dogleg (Newton)                 | Chol                   | 0.699 | 0.869   | 2400.3        | 156500 | 156813 | 156813 | 156500 | 50.48                |
| Dogleg (Newton)                 | GMW                    | 0.722 | 0.931   | 1024.0        | 16     | 20     | 20     | 13     | 1.46e <sup>-28</sup> |
| CG-Steihaug                     |                        | 0.722 | 0.931   | 1024.0        | 30     | 34     | 34     | 30     | 1.57e <sup>-28</sup> |
| Exact trust region <sup>c</sup> | Unmodified             | 0.722 | 0.931   | 1024.0        | 23     | 27     | 27     | 23     | 2.07e <sup>-28</sup> |
| Exact trust region              | Chol                   | 0.722 | 0.931   | 1024.0        | 23     | 27     | 27     | 23     | 2.07e <sup>-28</sup> |
| Exact trust region              | GMW                    | 0.722 | 0.931   | 1024.0        | 23     | 27     | 27     | 23     | 2.07e <sup>-28</sup> |
| Conjugate gradient algorithms   |                        |       |         |               |        |        |        |        |                      |
| Fletcher-Reeves                 | Back                   | 0.722 | 0.931   | 1024.0        | 2485   | 38830  | 2491   | 0      | 5.99e <sup>-17</sup> |
| Fletcher-Reeves                 | MT                     | 0.722 | 0.931   | 1024.0        | 82     | 412    | 412    | 0      | 4.47e <sup>-22</sup> |
| Polak-Ribière                   | Back                   | 0.722 | 0.931   | 1024.0        | 4135   | 64453  | 4145   | 0      | 8.80e <sup>-14</sup> |
| Polak-Ribière                   | MT                     | 0.722 | 0.931   | 1024.0        | 163    | 715    | 715    | 0      | 2.60e <sup>-21</sup> |
| Polak-Ribière +                 | Back                   | 0.722 | 0.931   | 1024.0        | 4135   | 64453  | 4145   | 0      | 8.80e <sup>-14</sup> |
| Polak-Ribière +                 | MT                     | 0.722 | 0.931   | 1024.0        | 163    | 715    | 715    | 0      | 2.60e <sup>-21</sup> |
| Hestenes-Stiefel                | Back                   | 0.722 | 0.931   | 1024.0        | 807    | 12741  | 813    | 0      | 1.32e <sup>-16</sup> |
| Hestenes-Stiefel                | MT                     | 0.722 | 0.931   | 1024.0        | 44     | 233    | 233    | 0      | 8.75e <sup>-23</sup> |
| Miscellaneous algorithms        |                        |       |         |               |        |        |        |        |                      |
| Simplex                         |                        | 0.722 | 0.931   | 1024.0        | 1701   | 2897   | 0      | 0      | 2.25e <sup>-28</sup> |
| Levenberg-Marquardt             |                        | 0.722 | 0.931   | 1024.0        | 112    | 118    | 62     | 0      | 1.38e <sup>-28</sup> |

<sup>a</sup> The algorithms auxiliary to the main optimisation algorithm include: Back – backtracking line search; MT – More and Thuente line search; Unmodified – no Hessian modification; Chol – Cholesky Hessian modification with added multiple of the identity matrix; GMW – Gill, Murray, and Wright modified Cholesky Hessian modification algorithm.

<sup>b</sup>  $i$  is the number of iterations and  $f$ ,  $g$ , and  $h$  are the number of function, gradient, and Hessian calls respectively.

<sup>c</sup> Algorithm failure due to the Hessian not being positive definite.



Table S6: Optimisation of the grid point  $S^2 = 0.970$ ,  $\tau_e = 2048$  ps, and  $R_{ex} = 0.149$  s<sup>-1</sup>.

|                         | $S^2$ | $\tau_e$ (ps) | $R_{ex}$ (s <sup>-1</sup> ) | $\chi^2$             |
|-------------------------|-------|---------------|-----------------------------|----------------------|
| Truth <sup>a</sup>      | 0.970 | 2048.00       | 0.149                       |                      |
| Modelfree4 <sup>b</sup> | 0.947 | 10000.00      | 0.000                       | 1.39                 |
| Dasha (LM) <sup>b</sup> | 1.000 | 57.12         | -1.00e <sup>-5</sup>        | 3.92                 |
| Dasha (NR) <sup>c</sup> | 0.970 | 2056.97       | 0.182                       | 4.87e <sup>-4</sup>  |
| relax <sup>d</sup>      | 0.970 | 2048.00       | 0.149                       | 7.30e <sup>-28</sup> |

<sup>a</sup>The true parameter values.

<sup>b</sup>Levenberg-Marquardt minimisation.

<sup>c</sup>Combined Newton-Raphson/conjugate gradient minimisation.

<sup>d</sup>Newton optimisation combined with the backtracking line search and the GMW Hessian modification.

Table S7: Optimisation of the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 32$  ps.

|                         | $S_f^2$ | $S_s^2$ | $\tau_s$ (ps) | $\chi^2$             |
|-------------------------|---------|---------|---------------|----------------------|
| Truth <sup>a</sup>      | 0.952   | 0.582   | 32.00         |                      |
| Modelfree4 <sup>b</sup> | 0.622   | 0.892   | 288.93        | 3.47                 |
| Dasha (LM) <sup>b</sup> | 1.000   | 0.555   | 28.53         | 5.68e <sup>-3</sup>  |
| Dasha (NR) <sup>c</sup> | 0.803   | 0.690   | 52.10         | 9.15e <sup>-3</sup>  |
| relax <sup>d</sup>      | 0.952   | 0.582   | 32.00         | 3.31e <sup>-28</sup> |

<sup>a</sup>The true parameter values.

<sup>b</sup>Levenberg-Marquardt minimisation.

<sup>c</sup>Combined Newton-Raphson/conjugate gradient minimisation.

<sup>d</sup>Newton optimisation combined with the backtracking line search and the GMW Hessian modification.

Table S8: High precision optimisation of the grid point  $S_f^2 = 0.952$ ,  $S_s^2 = 0.582$ , and  $\tau_s = 32$  ps after modification and recompilation of the Modelfree4 and Dasha source code.

|                                    | $S_f^2$ | $S_s^2$ | $\tau_s$ (ps) | $\chi^2$            |
|------------------------------------|---------|---------|---------------|---------------------|
| Truth <sup>a</sup>                 | 0.952   | 0.582   | 32.00         |                     |
| Modelfree4 (HP) <sup>b</sup>       | 0.622   | 0.892   | 288.93        | 3.47                |
| Modelfree4 (Debugged) <sup>c</sup> | 0.730   | 0.758   | 75.39         | 5.8e <sup>-3</sup>  |
| Modelfree4 (Both) <sup>d</sup>     | 0.900   | 0.616   | 36.94         | 1.0e <sup>-4</sup>  |
| Dasha (LM+HP) <sup>e</sup>         | 0.952   | 0.582   | 32.00         | 2.08e <sup>-7</sup> |
| Dasha (NR+HP) <sup>f</sup>         | 0.911   | 0.608   | 35.75         | 2.06e <sup>-5</sup> |

<sup>a</sup>The true parameter values.

<sup>b</sup>HP – high precision optimisation where the chi-squared cutoff is 1e<sup>-25</sup>, the maximum number of iterations is either 1e<sup>7</sup> or infinite, and no gradient or step length cutoffs are used.

<sup>c</sup>Removal of the bug in the Levenberg-Marquardt algorithm.

<sup>d</sup>Both high precision and debugged optimisation.

<sup>e</sup>High precision Levenberg-Marquardt minimisation.

<sup>f</sup>High precision combined Newton-Raphson/conjugate gradient minimisation.

Table S9: Optimisation of the grid point  $S^2 = 0.388$ ,  $\tau_e = 128$  ps, and  $R_{ex} = 0.223$  s<sup>-1</sup>.

|                         | $S^2$ | $\tau_e$ (ps) | $R_{ex}$ (s <sup>-1</sup> ) | $\chi^2$             |
|-------------------------|-------|---------------|-----------------------------|----------------------|
| Truth <sup>a</sup>      | 0.388 | 128.000       | 0.223                       |                      |
| Modelfree4 <sup>b</sup> | 0.263 | 526.316       | 1.053                       | 1371.79              |
| Dasha (LM) <sup>b</sup> | 0.388 | 128.270       | 0.231                       | 1.28e <sup>-4</sup>  |
| Dasha (NR) <sup>c</sup> | 0.388 | 128.270       | 0.231                       | 1.28e <sup>-4</sup>  |
| relax <sup>d</sup>      | 0.388 | 128.000       | 0.223                       | 1.25e <sup>-28</sup> |

<sup>a</sup>The true parameter values.

<sup>b</sup>Levenberg-Marquardt minimisation.

<sup>c</sup>Combined Newton-Raphson/conjugate gradient minimisation.

<sup>d</sup>Newton optimisation combined with the backtracking line search and the GMW Hessian modification.

Table S10: The parameter values of the optimised diffusion models of cytochrome c<sub>2</sub>.

| Diffusion model     | $\mathfrak{D}$ (1e <sup>7</sup> s <sup>-1</sup> ) <sup>a</sup> |                  |                  | $\mathfrak{D}$ (deg) <sup>a</sup> |                     |          | k <sup>b</sup> | $\chi^2(\hat{\theta})^c$ | AIC <sup>d</sup><br>( $\chi^2(\hat{\theta}) + 2k$ ) |
|---------------------|----------------------------------------------------------------|------------------|------------------|-----------------------------------|---------------------|----------|----------------|--------------------------|-----------------------------------------------------|
|                     | $\mathfrak{D}_x$                                               | $\mathfrak{D}_y$ | $\mathfrak{D}_z$ | $\phi$ or $\alpha$                | $\theta$ or $\beta$ | $\gamma$ |                |                          |                                                     |
| Tensor <sup>e</sup> |                                                                |                  |                  |                                   |                     |          |                |                          |                                                     |
| Sphere              | 1.630                                                          | 1.630            | 1.630            |                                   |                     |          | 1              | 53.2                     |                                                     |
| Prolate spheroid    | 1.438                                                          | 1.438            | 1.928            | 78.0                              | 72.5                |          | 4              | 25.1                     |                                                     |
| Oblate spheroid     | 1.700                                                          | 1.700            | 1.432            | -30.0                             | 20.9                |          | 4              | 24.6                     |                                                     |
| Ellipsoid           | 1.405                                                          | 1.566            | 1.829            | 19.2                              | -27.8               | -33.5    | 6              | 21.4                     |                                                     |
| relax               |                                                                |                  |                  |                                   |                     |          |                |                          |                                                     |
| Sphere              | 1.624                                                          | 1.624            | 1.624            |                                   |                     |          | 45             | 95.3                     | 185.3                                               |
| Prolate spheroid    | 1.443                                                          | 1.443            | 1.913            | 162.0                             | 101.7               |          | 48             | 44.0                     | 140.0                                               |
| Oblate spheroid     | 1.694                                                          | 1.694            | 1.434            | 108.7                             | 30.3                |          | 48             | 44.3                     | 140.3                                               |
| Ellipsoid           | 1.415                                                          | 1.559            | 1.826            | 155.6                             | 104.4               | 13.6     | 50             | 38.6                     | 138.5                                               |

<sup>a</sup>As the diffusion tensor eigenvalues for all diffusion models within Tensor are defined as  $\mathfrak{D}_{xx} \geq \mathfrak{D}_{yy} \geq \mathfrak{D}_{zz}$ , the eigenvalues have been rearranged to match the convention used here. The orientational diffusion parameter set  $\mathfrak{D}$  consists of the parameters  $\{\theta, \phi\}$  for the oblate and prolate spheroids and  $\{\alpha, \beta, \gamma\}$  for the ellipsoid. The orientational parameters cannot be compared because of three differences between Tensor and relax. Firstly, within Tensor all eigenvalues are defined strictly as  $\mathfrak{D}_x \geq \mathfrak{D}_y \geq \mathfrak{D}_z$ . As the ellipsoid eigenvalues are defined as  $\mathfrak{D}_x \leq \mathfrak{D}_y \leq \mathfrak{D}_z$  in this paper the two ellipsoids experience a frame shift requiring the rotation of all three Euler angles. For both the prolate and oblate spheroids defined in this paper the unique axis of the tensor is set to  $\mathfrak{D}_z$ . In Tensor, because of the retention of the above constraints, the unique axes of the prolate and oblate spheroids are  $\mathfrak{D}_x$  and  $\mathfrak{D}_z$  respectively. Hence both spheroids also exhibit frame shifts. In addition, the Euler angle and spherical angle conventions in Tensor are unknown. Finally it is unknown how Tensor handles the glide reflection and translation symmetries of the orientational parameter set.

<sup>b</sup>The number of parameters in the model.

<sup>c</sup>The chi-squared values between Tensor and relax cannot be compared as they are defined differently.

<sup>d</sup>The chi-squared value from Tensor cannot be used to calculate the AIC value as derived in (d'Auvergne and Gooley, 2003). The model which best fits the relaxation data is that with the lowest criterion value.

<sup>e</sup>The results presented here are taken from Table 4 of (Cordier et al., 1998).

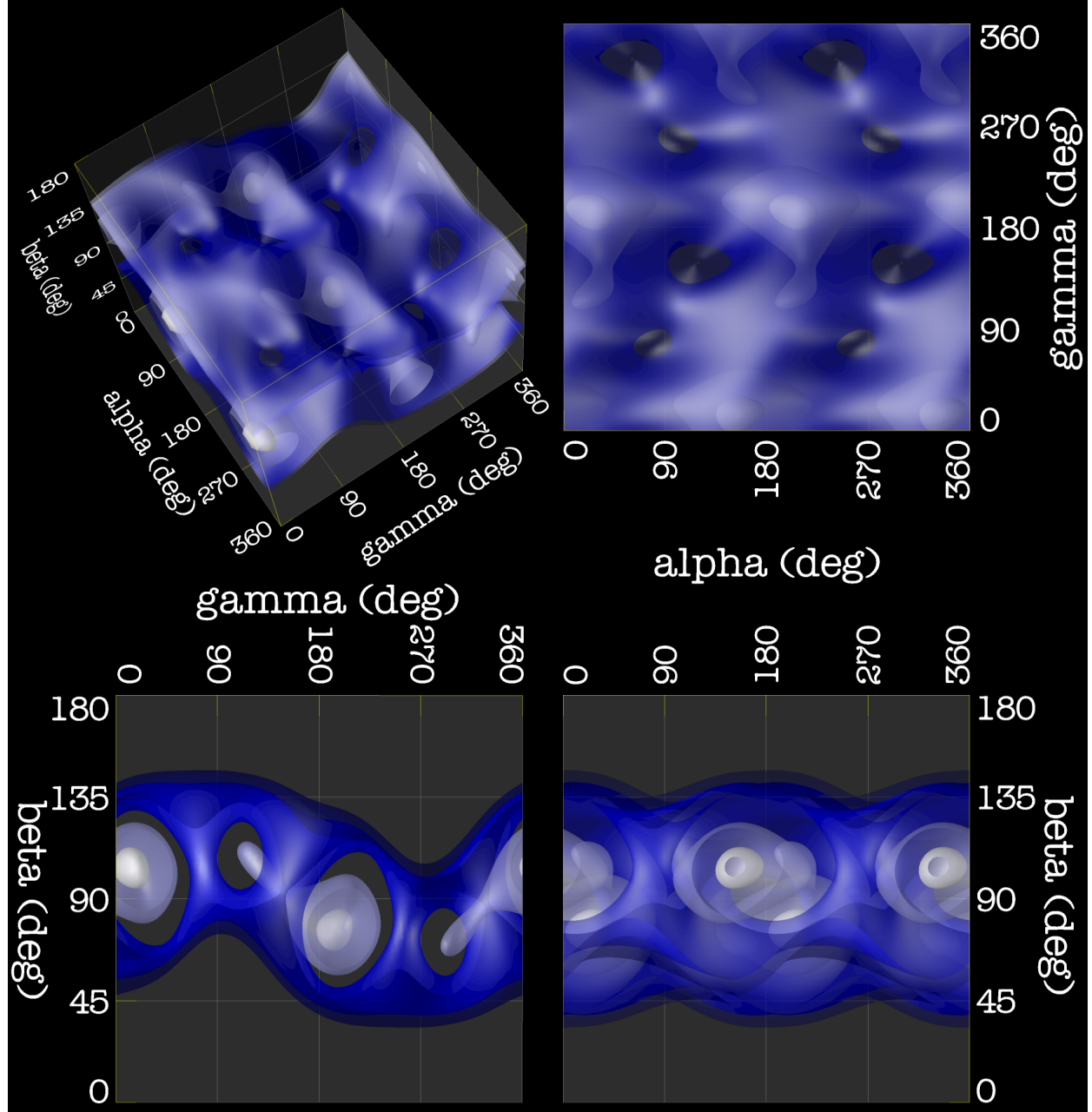


Figure S24: Map of the chi-squared space of the ellipsoid orientational diffusion tensor parameters  $\mathfrak{D} = \{\alpha, \beta, \gamma\}$  of cytochrome  $c_2$ . This figure demonstrates the four identical minima of the space due to both the translational and glide reflection symmetries of the ellipsoid orientational parameter set  $\mathfrak{D}$  arising from the squared direction cosines in the weights of the rotational correlation function. The translational symmetry is visible as the subspace between  $\pi \leq \alpha \leq 2\pi$  being a perfect duplication of the subspace between  $0 \leq \alpha \leq \pi$ . The glide reflection can be seen as the subspace between  $\pi \leq \gamma \leq 2\pi$  being a duplication of the subspace between  $0 \leq \gamma \leq \pi$  where two mirror image reflections have occurred about both  $\alpha$  and  $\beta$ . The  $\chi^2$  values of the four isosurfaces from outermost to innermost are 300, 250, 100, and 50 respectively (dark blue to white). The resolution of the plot is 100 data points per dimension.

## References

- Broyden, C. G.: 1970, 'The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations'. *J. Inst. Maths. Applics.* **6**(1), 76–90.
- Clore, G. M., A. Szabo, A. Bax, L. E. Kay, P. C. Driscoll, and A. M. Gronenborn: 1990, 'Deviations from the simple 2-parameter model-free approach to the interpretation of N-15 nuclear magnetic-relaxation of proteins'. *J. Am. Chem. Soc.* **112**(12), 4989–4991.
- Cordier, F., M. Caffrey, B. Brutscher, M. A. Cusanovich, D. Marion, and M. Blackledge: 1998, 'Solution structure, rotational diffusion anisotropy and local backbone dynamics of *Rhodobacter capsulatus* cytochrome c2.'. *J. Mol. Biol.* **281**(2), 341–361.
- d'Auvergne, E. J. and P. R. Gooley: 2003, 'The use of model selection in the model-free analysis of protein dynamics.'. *J. Biomol. NMR* **25**(1), 25–39.
- Fletcher, R.: 1970, 'A new approach to variable metric algorithms'. *Comp. J.* **13**(3), 317–322.
- Fletcher, R. and C. M. Reeves: 1964, 'Function minimization by conjugate gradients'. *Comp. J.* **7**(2), 149–154.
- Gill, P. E., W. Murray, and M. H. Wright: 1981, *Practical Optimization*. Academic Press.
- Goldfarb, D.: 1970, 'A family of variable-metric methods derived by variational means'. *Math. Comp.* **24**(109), 23–26.
- Hestenes, M. R. and E. Stiefel: 1952, 'Methods of conjugate gradients for solving linear systems'. *J. Res. Natn. Bur. Stand.* **49**(6), 409–436.
- Levenberg, K.: 1944, 'A method for the solution of certain non-linear problems in least squares'. *Quarterly of Applied Mathematics* **2**, 164–168.
- Marquardt, D. W.: 1963, 'An algorithm for least squares estimation of non-linear parameters'. *SIAM J.* **11**, 431–441.
- Moré, J. J. and D. J. Thuente: 1994, 'Line search algorithms with guaranteed sufficient decrease'. *ACM Trans. Maths. Softw.* **20**(3), 286–307.
- Nocedal, J. and S. J. Wright: 1999, *Numerical Optimization*, Springer Series in Operations Research. New York: Springer-Verlag.
- Polak, E. and G. Ribière: 1969, 'Note sur la convergence de méthodes de directions conjuguées'. *Revue Française d'Informatique et de Recherche Opérationnelle* **16**, 35–43.
- Shanno, D. F.: 1970, 'Conditioning of quasi-Newton methods for function minimization'. *Math. Comp.* **24**(111), 647–656.
- Steihaug, T.: 1983, 'The conjugate gradient method and trust regions in large scale optimization'. *SIAM J. Numer. Anal.* **20**(3), 626–637.